Automated Optimization of Broiler Production

Johansen, Simon Vestergaard

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

AUTOMATED OPTIMIZATION OF BROILER PRODUCTION

BY
SIMON VESTERGAARD JOHANSEN

DISSERTATION SUBMITTED 2019

AALBORG UNIVERSITY
DENMARK

Automated Optimization of Broiler Production

Ph.D. Dissertation

Simon Vestergaard Johansen

Dissertation submitted September, 2019

Dissertation submitted: September, 2019

PhD supervisor: Assoc. Prof. Jan D. Bendtsen
Aalborg University

PhD Supervisor (industrial): Jesper Mogensen
SKOV A/S

PhD committee: Associate Professor Anders la Cour-Harbo (chairman)
Aalborg University

Professor Daniel Bercknans
KU Leuven

Head of Group R&D Martin Simonsen
SKIOLD A/S

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Electronic Systems

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-511-6

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2019

Abstract

Broiler (chicken for meat production) growth optimization reduces the amount of feed, water and electricity required to produce a mature broiler. It is important, as the global poultry meat consumption of 120×10^9 kg per year is predicted to increase by around 2% per year, where broiler meat represents the majority. The main objective of *state-of-the-art* industrial broiler production is to facilitate efficient maturation of broilers under ethically acceptable conditions at scale, e.g., 40.000 broilers per batch. The primary expense in broiler production is feed, and subsequently the primary key performance measure is feed conversion ratio (FCR), i.e., feed efficiency. Existing scientific control literature is incompatible with *state-of-the-art* industrial broiler production, as it exclusively focuses on active feed control despite the fact that unrestricted feeding regimes unanimously are used in practice. Climate conditions, e.g., temperature and light intensity, can be controlled and are known to significantly influence broiler growth, effectively providing a potential avenue for growth optimization, and will thus be the subject of this study.

Data driven modeling techniques are used given the absence of first-principle mathematical broiler growth models explaining the temporal relationship between broiler climate conditions and broiler growth. A sufficiently accurate dynamic model is required for model based controller synthesis in general, and broiler growth optimization is no exception. Dynamic neural network (DNN) models are successfully trained on industrial scale production data. Ensemble DNN models are successfully used to forecast broiler weight on ≈4 years of production data using environmental broiler house variables. A mutual information based input variable selection algorithm is used to automatically determine the DNN input-output structure. Evidence for the environmental broiler house variables having a significant dynamic influence on broiler growth, feed and drinking behavior was established.

Iterative learning control (ILC) provides a potential solution given the repeated nature of the production process, as it has been especially developed for systems that make repeated executions of the same finite duration task. An ILC algorithm in conjunction with the DNN model resulted in a normalized FCR reduction of 2.4% over 2 batches. A modified terminal ILC algorithm resulted in a FCR reduction of 1.4% over a single batch, and is modified to gradually minimize the terminal FCR and better cope with the uncertain na-

ture of the data driven model. A heuristic broiler growth model is formulated to assist broiler optimization development in simulation and is used to demonstrate that the modified terminal ILC works in simulation, and that an iterative search strategy, like ILC, is required for FCR minimization. A combined FCR improvement of 3.8% was obtained for the two FCR optimization attempts – corresponding to a yearly saving of 8.500€ and 26.7 tones of broiler feed per broiler house.

Resumé

Vækstoptimering af industriel slagtekyllingeproduktion reducerer den påkrævede mængde foder, vand og elektricitet der skal til for at producere en slagteklar kylling. Det er et vigtigt taget i betragtning af, at der globalt bliver spist 120×10^9 kg kyllingekød per år med en forventet stigning på omkring 2% per år. Hovedformålet med moderne industriel slagtekyllingeproduktion er at faciliterer effektiv modning af slagtekyllinger under etisk acceptable forhold på en stor skala, for eksempel 40.000 kyllinger per hold. Hovedudgiften for slagtekyllingeproduktion er foder, hvoraf det primære nøgletal er foderkonverteringsraten (FKR), altså foderudnyttelse. Den eksisterende videnskabelige litteratur er ikke kompatibel med moderne slagtekyllingeproduktion, fordi den eksklusivt bruger aktiv foderregulering selv om kyllingerne har ubegrænset adgang til foder i praksis. Klima, eksempelvis temperatur og lysintensitet, kan blive reguleret og har en signifikant indflydelse på vækst, hvilket giver en potentiel vækstoptimeringsmulighed og derfor er hovedemnet for dette studie.

Datadrevne modelleringsteknikker bliver taget i brug da første princips slagtekyllingevækstmodeller ikke eksisterer, som forklarer det dynamiske forhold mellem slagtekyllingevækst og klima. En tilstrækkelig præcis dynamisk model er generelt påkrævet til syntetisering af en modelbaseret regulator, hvilket også er tilfældet for slagtekyllingevækstoptimering. Dynamiske neurale netværk (DNN) modeller bliver succesfuldt trænet på industriel skala produktionsdata. Ensemble DNN modeller er succesfuldt brugt til at forudse kyllingevægt på ≈4 års productionsdata ved brug af staldmiljø variable. En fælles informations baseret input variabel selektering er succesfuldt brugt til at bestemme DNN modellens input-output struktur. Beviser for at staldmiljøet har en signifikant dynamisk indflydelse for kyllingetilvækst, foder- og drikkeadfærd er fastlagt.

Iterativ lægnings regulering (ILR) udgør en potentiel løsning taget den gentagende natur af slagtekyllingeproduktionsprocessen, da metoden er specielt udviklet til at håndterer systemer der gentager tidsbegrænsede opgaver. En ILR algoritme i samspil med DNN modellen resulterede i en normaliseret FKR reduktion på 2.4% over 2 hold. En modificeret terminal ILR algoritme resulterede i en ILR reduktion på 1.4% over et enkelt hold, og er modificeret til gradvist at minimerer terminal FKR og bedre håndterer den usikre natur forsaget af den data drevne model. En heuristisk slagtekyllingevækstmodel er formuleret til at assisterer slagtekyllingevækstoptimeringsudvikling og er brugt til at demon-

strere den modificeret terminal ILR algoritme i et simuleringsmiljø. En iterativ søgestrategi, som ILR, er påkrævet for FKR minimering. En kombineret ILR forbedring på 3.8% blev opnået for de to ILR optimeringsforsøg – svarende til en årlig besparelse på 8.500€ og 26.7 tons kyllingefoder per hus.

Contents

Acknowledgments

I'm grateful to SKOV for the opportunity to be a pioneer in broiler growth optimization research. Throughout this study, I have come to regard broiler production as a globally significant field that receives a disproportionately small level of attention from my technically academic peers compared to its sheer scale. Maybe this is partly due to the silly stigma surrounding chickens, the degree of open-mindedness of the typical technical scientist, a lack of branding, or my own malleable personal biases.

I wish to thank my colleagues at SKOV, most notably team FarmOnline and the steering committee, for their continued support and many insightful discussions. In particular, Jesper Mogensen and Martin Riisgaard-Jensen for sharing their industrial perspectives, Ebbe Kronborg for his honesty and keen eye for details, Broiler Application Expert Martin Riishøj Jensen for sharing his insights, and finally Søren Hørup for his continued challenge of my academic ideas and almost infinite amount of theories about how the world "should" work.

I wish to express my sincere gratitude to my supervisor Jan Bendtsen at Aalborg University for providing excellent guidance. I am genuinely thankful for the hospitality and supervision extended to me by Eric Rogers and Bing Chu at University of Southampton in England. Without the combination of outside rigorous academic and industrial perspectives, the heuristic broiler model and subsequent simulation study might not have existed today.

Animal welfare is a complicated subject that requires attention to many levels of analysis with many competing points of view, ranging from animal rights activists to industrial integrators. This thesis fails at maintaining a neutral perspective due to its inherent industrial interests. However, it is the authors understanding that the developed research is equally capable of solving animal welfare problems, thus presenting a balanced outlook of the future. *May all beings everywhere be happy and free, and may the actions inspired by this research be carried out in good spirit and contribute in some way to that happiness and to that freedom for all.*

Simon Vestergaard Johansen
Aalborg University, Monday 30$^{\text{th}}$ September, 2019

Part I

Summary

1 Introduction

This chapter provides the necessary background and motivation for studying automatic optimization of broiler batch production.

1.1 Motivation

Due to the growing middle class in many developing countries, the global meat consumption is predicted to continue rising in the foreseeable future [OECD, 2015]. The highest growth is predicted in poultry meat, which has shown a steady increase in total consumption and consumption per capita in recent years, as depicted on Figure 1.1. However, the consumption per capita is predicted to level out despite the growing total consumption. Recent yearly average production increase estimates range from 1.6% [OECD, 2017, pp. 131] to 2.4% [OECD, 2015, pp. 136].

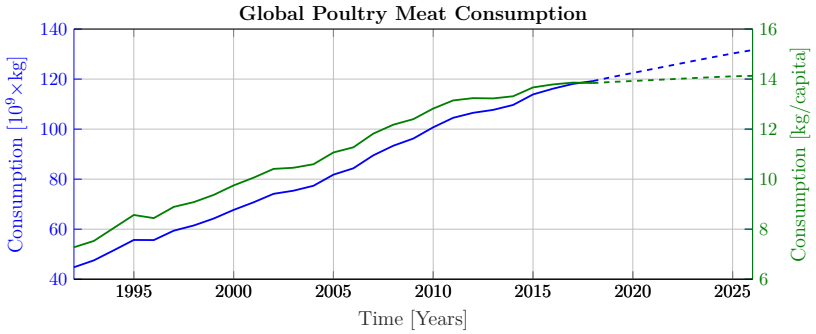


Figure 1.1: Global poultry meat consumption with dashed 10 year ahead projections[OECD, 2018b].

Broiler (chicken for meat production) meat represents the majority of global poultry production – broiler meat represents 83% of poultry meat in United states of America [National Chicken Council, 2018a], which is the worlds largest broiler meat producer. One reason being that broiler production has the highest yield per feed unit among land animals, making chicken meat a relatively inexpensive source of animal protein. Broiler growth maximization reduces the amount of feed, water and electricity required to produce a mature broiler.

To meet the growing broiler meat consumption demand, the broiler meat supply chain must be up for the task. A simplified representation is depicted on Figure 1.2, where:

- **Feed Mills** supply the growers with feed.
- **Breeders** supply the hatcheries with eggs using laying hens.

- **Hatcheries** hatch the eggs and supply growers with day old broilers.
- **Growers** supply the slaughterhouses with mature broilers by maturing the day old broilers.
- **Slaughterhouses** produce broiler meat from mature broilers.

Hence, broiler production is not a single cohesive process, but rather a series of dependent processes.

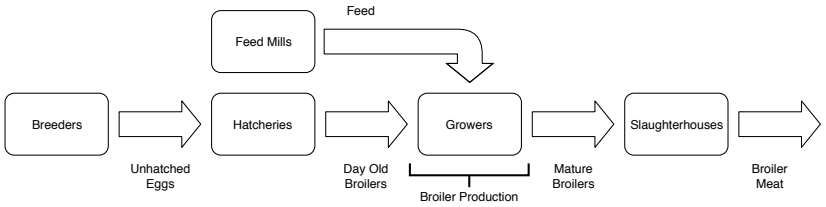


Figure 1.2: Simplified broiler supply chain overview.

The main objective of industrial scale broiler production is to facilitate efficient maturation of broilers under ethically acceptable conditions. The primary expense in broiler production is feed, and subsequently the primary key performance measure is feed conversion ratio (FCR) defined by

$$\text{FCR} = \frac{\text{Feed in kg}}{\text{Live weight in kg}}. \tag{1.1}$$

A typical FCR at day 34 under ideal laboratory conditions for the most popular fast growing broiler strains are 1.53 for ROSS 308[Aviagen, 2014a] and 1.48 for COBB 500[Cobb-Vantress, 2018] – see Figure 1.3 for growth information throughout the growing period for these strains. Faster growth most notably reduces the required growing time, which subsequently reduces the cumulative metabolic maintenance cost of organic material.

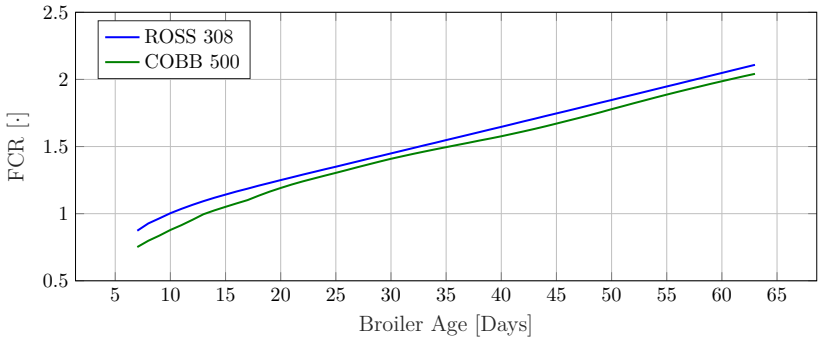


Figure 1.3: FCR performance by day of popular fast growing broiler strains. [Aviagen, 2014a, pp. 3][Cobb-Vantress, 2018, pp. 21].

Artificial selection of broilers has received tremendous attention and has resulted in fast growing strains with a profound increase in both growth rate and feed efficiency. This has resulted in both a 50% reduction in FCR and over 400% increase in broiler growth rate between 1957 and 2005 [Zuidhof et al., 2014], and a reduction in time to reach 1500g of live weight from 120 days in 1925 to 30 days in 2005[Bessei, 2006]. Optimizing environmental factors, such as providing the optimal climate and nutrition, are paramount to properly express this genetic potential. For reference, the average FCR in USA is 1.83[National Chicken Council, 2018b] that, if naively assuming only 34 day old ROSS 308 broiler are grown, represents an average potential FCR gap of more than 0.3 – corresponding to 30%! It is believed that this gap is caused by suboptimal management, and that local optimization helps to bridge this gap, which will be the subject of this study.

For simplicity, broiler production will henceforth refer to the maturation of day old broilers at the growers, as indicated on Figure 1.2.

1.2 Broiler Production

Broiler production is the process of growing day old broiler chicks from the hatcheries, as depicted on Figure 1.4a, into mature slaughter ready broilers, as depicted on Figure 1.4b. This is carried out at scale in dedicated broiler houses as depicted on Figure 1.5, each containing 40-50,000 broilers, equipped with climate and light control in addition to ad Librium supply of both feed and water. It typically only takes 34 days to grow a 2050g slaughter ready ROSS 308 broiler.

(a) Day old broiler. **(b)** Mature broiler.

Figure 1.4: Broiler production.

Figure 1.5: Example of industrial scale broiler house.

1.2.1 Feeder and Drinkers

Feed and water is provided ad Libitum to encourage maximal growth, and is readily available through multiple parallel drinker and feeder "lines". Figure 1.7b depicts a broiler drinking from a nippler, where water is emitted when a metal peg is displaced – Figure 1.7a depicts a nippler water line. Figure 1.7c depicts a pan style feeder line, where the pan is automatically refilled with feed through the attached rod. Feeder and drinker lines are often positioned parallel to each other as depicted on Figure 1.7d.

1.2.2 Monitoring

Various sensors continuously monitor and log the production and are remotely accessible through the Internet. This includes the broiler weight that is measured using one or more weighing pads as depicted on Figure 1.8. The cumulative feed and water consumption are automatically measured using water meters and feed weights. Air quality related sensors, like temperature, humidity and CO2, are positioned at key locations close to ground level as depicted on Figure 1.6.

Figure 1.6: Combined temperature and humidity sensor.

(a) Example of nippler style water line.

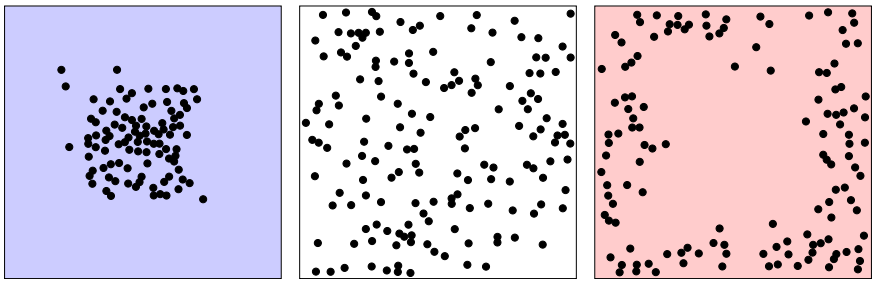
(b) Broiler drinking.

(c) Example of pan style feeder line.

(d) Parallel feeder (yellow; bottom left) and drinker (red; top right) lines.

Figure 1.7: Feeder and drinker line examples.

Figure 1.8: Different weighing pad types.



(a) Cold. **(b)** Thermoneutral. **(c)** Hot.

Figure 1.9: Broiler positioning behavior at different temperatures. A black dot denotes a broiler and the outer black box of each illustration denotes the broiler house walls [Aviagen, 2014b, pp. 21-22].

1.2.3 Climate Control

Proper climate control is a crucial factor for high production efficiency, as previously mentioned. The correct temperature depends on how developed the broilers are, of which different temperature states influence the broilers as follows:

Cold Broilers tend to cluster together to preserve heat at colder areas if they are too cold – typically *away* from the house walls as depicted on Figure 1.9a. The temperature along the walls are typically colder, if the outside temperature is lower than the inside temperature.

Thermoneutral Broilers tend to be evenly distributed across the house floor as depicted on Figure 1.9b.

Hot Broilers tend to cluster together at colder production areas if they are too hot – typically *along* the house walls as depicted on Figure 1.9c. Broilers pant to cool down, effectively increasing evaporation of water from the lunges and throat, which requires additional energy and subsequently lowers feed efficiency.

There exist a very narrow optimal temperature range where growth is maximized regardless of how developed the broilers are , i.e., the difference between energy intake and expenditure is maximized. This is depicted on Figure 1.10, where deviations from the optimal temperature results in decreased feed efficiency. Likewise, for sufficiently cold or hot temperatures the broilers can lose weight due to a total energy deficit, as depicted on Figure 1.10 and might be the case for Figure 1.9a and Figure 1.9c.

Broilers gradually gain the ability to self-regulate body temperature around day 9. Feathering typically starts around day 6 and ends around day 25, which drastically increases cold resistance. This makes either too hot or cold temperatures particularly detrimental for younger broilers, which responds by limiting

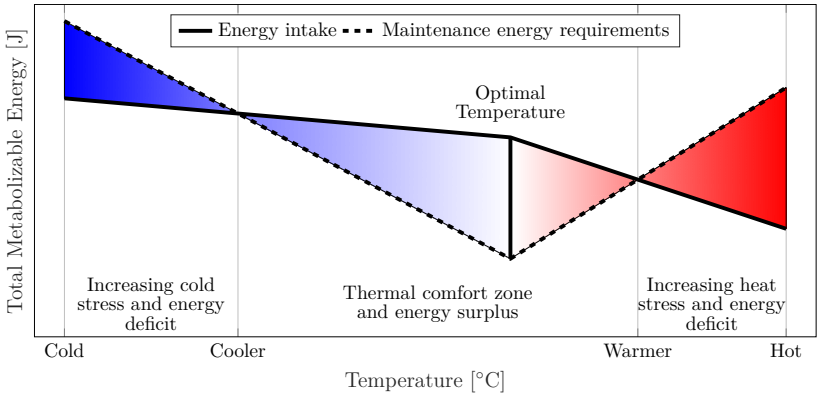


Figure 1.10: Total metabolizable energy for different temperature categories in terms of energy intake and maintenance energy requirements. Blue denotes a cold temperature, red denotes hot temperature, and white denotes thermoneutral temperature. The optimal temperature is marked with a vertical line [Aviagen, 2010, pp. 4].

eating and drinking behavior. Furthermore, the recommended temperature profiles decrease almost linearly with age as depicted on Figure 1.11. The heating requirements are high in the beginning of the batch, as the biological heat producing mass is low and the required temperature is high, while ventilation becomes more prominent later in the batch, where the biological heat producing mass is high and the required temperature is low. Note that the experienced temperature of the broiler depends on air temperature, relative humidity and wind speed.

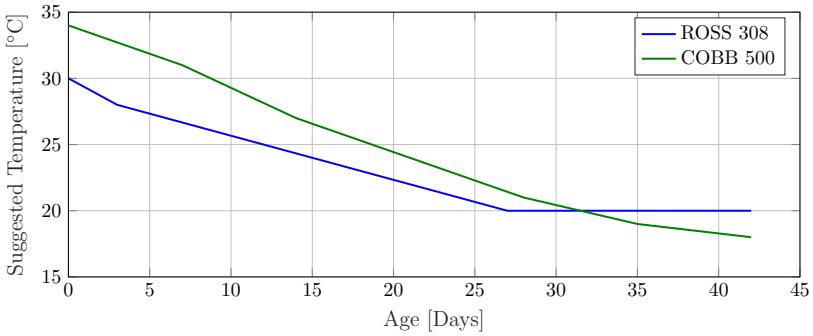


Figure 1.11: Suggested temperature profiles for ROSS 308[Aviagen, 2014b, pp. 16] and COBB 500[Cobb-Vantress, 2018, pp. 21] (assuming 30 week or older parent stock) by the manufacturer.

In temperate climates, like in Denmark, the broiler house climate is typically controlled by pulling in cold air from the outside through side inlets and out through exhausts vents in the ceiling, and heating the air using heaters. The

air inlets are typically positioned lengthwise the broiler house as depicted on Figure 1.12a, and the degree of openness is controlled by pulling a cord using a linear actuator. Figure 1.12b and Figure 1.12c depicts the air inlets in an open and closed position, and the exhaust vents can be depicted on the ceiling of Figure 1.5. Heater lines are typically positioned below the air inlet line to heat incoming air as depicted on Figure 1.12c, and is filled with circulating water heated by a furnace. Higher cooling capacity solutions are often required in tropical climates that involve large fans, cooling pads with running water and misters.

(a) Side air inlets. **(b)** Open inlet. **(c)** Closed inlet with heater.

Figure 1.12: Examples of air inlets and heater.

1.2.4 Automatic Broiler Production Optimization

Broiler production can be regarded as a system of considerable complexity that the farmer must navigate – especially considering that each broiler house is unique and requires individual assessment. It is up to the farmer to control the air quality and temperature, feed mix and lighting, among others – despite changing outside conditions, feed quality and external stress factors, e.g., noise pollution. These are all crucial and interconnected factors for the broiler growth process. The experienced farmer uses the previously mentioned information and experience to make smarter and more deliberate decisions, tailored to each batch – similar to a control system, as depicted on the top of Figure 1.13.

However, the inexperienced farmer is not as attuned to this framework, resulting in suboptimal broiler management and production efficiency. Currently, the hatcheries supplying the farmer with broilers for broiler production only provide vague guidelines for dealing with problems – for instance, “if there is a high early mortality rate”, then a possible cause is “Poor chick quality” and the recommended action is to “Check hatchery practice and egg hygiene”[Aviagen, 2014b]. Alternatively, broiler application experts can be hired to supervise and

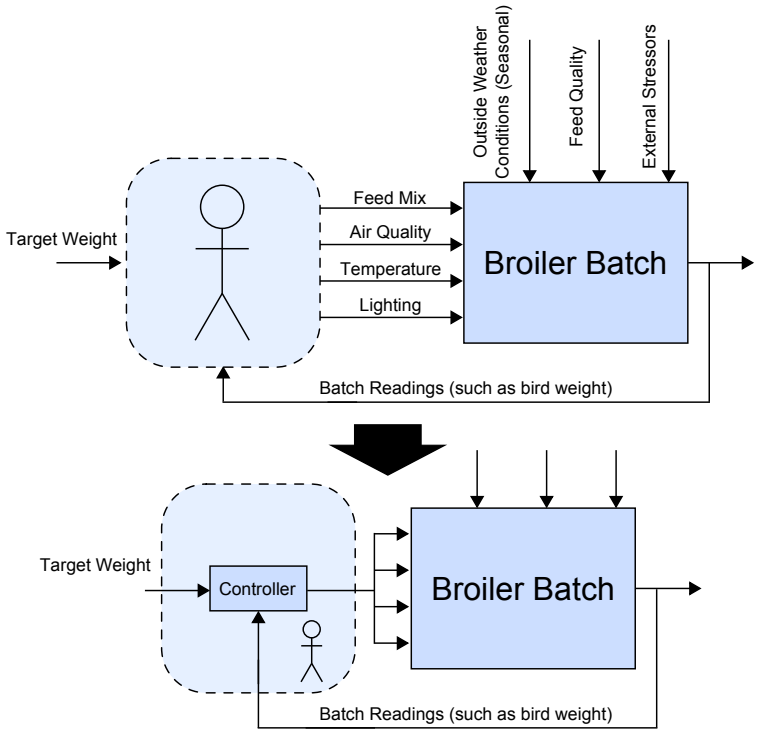


Figure 1.13: Automatic Broiler Production Optimization. Top: Current manual control. Bottom: Proposed automated control system.

manage the production, which are not always practically feasible at scale, as such experts are quite rare.

This study is initiated by SKOV A/S, which is an industry leader in the international market for climate control and production monitoring of broiler production. The vision of this study is to enable the equipment and systems supplied by SKOV A/S to assist the personnel responsible for the production to perform the previously mentioned adjustments automatically *at scale*. By using already available readings described in Section 1.2.2, the intention is to assist the farmer as depicted on the bottom of Figure 1.13 by "closing the loop". This can be carried out by either literally closing the loop or using a Human-In-The-Loop controller, where the farmer must approve of the controllers actions, as live animals are involved. In both cases the primary objective of this study is to assist the inexperienced farmer by extracting "expert knowledge" from previous batches and applying it to the current batch – similar to the experienced farmer.

1.3 Research Objectives

The primary objective of this study is to both minimize the environmental impact and maximize the profit margin of broiler production though suitably designed climate control. Specifically, to implement an "automatic farmer assistant" that gradually learns the batch production process, enabling it to improve production performance over time. For this purpose the following research objectives must be meet:

Objective 1: Develop a data driven dynamic broiler growth model. Data driven modeling techniques are used given the absence of first-principle mathematical broiler growth models explaining the temporal relationship between broiler climate conditions and broiler growth. A sufficiently accurate dynamic model is required for model based controller synthesis in general, and broiler growth optimization is no exception.

Such a model has the potential for aiding broiler site management by, e.g., forecasting future broiler weight that allows for timely intervention and enabling knowledge transfer between broiler houses. However, the existence of a dynamic model most notably allows advanced control theory techniques to be introduced in broiler production, such as fault tolerant detection and control.

Objective 2: Broiler production optimization using Iterative Learning Control. Iterative Learning Control (ILC) is an advanced control technique that gradually learns and improves performance of repetitive processes iteration by iteration with high precision. Existing ILC algorithms are applied to broiler production, and new theory will be developed to the extent needed. Given the practical and industrial applications of this study, the developed algorithm has to be tested on actual *state-of-the-art* industrial scale broiler productions.

2 State of the Art

This chapter provides an overview of related work, followed by a description of how the contributions of this study extend it.

2.1 Biological Broiler Research

This section provides a non-comprehensive overview of biological broiler research from the perspective of the grower of the broiler supply chain introduced in Section 1.1.

2.1.1 Temperature

Broilers live comfortably only in a fairly narrow temperature range, of which low and high temperatures are stressful and negatively impact growth performance [Olfati et al., 2018; Yunianto et al., 1997]. [Manning and Wyatt, 1990] found that broilers acclimate easier to lower temperatures, which help explain that heat stress receives significantly more academic attention than cold stress. See [Lin et al., 2006] for a review of strategies for preventing heat stress in broilers. See [Farag and Alagawany, 2018] for an overview of the physiological response to heat stress.

[Olfati et al., 2018] found that heat or cold stressed broilers had decreased growth performance. [May and Lott, 2000] investigated the impact of different temperature profiles on weight gain and feed conversion ratio in a laboratory setting. The investigated temperature profiles starts at $\{28, 29, 30, 31, 32\}$ °C at day 1 and decline by 0.3 °C per day, but found no statistically significant weight and FCR difference between the groups – only a high early mortality for starting temperature of 28 °C. Due to changing production conditions and improved genetics, [Cassuce et al., 2013] proposed an updated optimal temperature strategy by analyzing the response of starting temperature at $\{27, 30, 33, 36, 39\}$ °C at day 1, and declining 3 °C per week for a total of 3 weeks. Using regression formulas, the optimal temperature strategy was found to equal [31.3, 26.3, 22.5–23.2] °C for the first day of week 1 to 3, and linearly interpolated between the weeks. [Cândido et al., 2016] found that a mildly cold thermal environment, comprising of [27, 24, 21] °C for week 1 to 3 with FCR = 1.20, outperformed both the "standard" temperature profile with FCR = 1.27, and [Cassuce et al., 2013] with FCR = 1.26. [Henriksen et al., 2016], among others, studied the impact of increasing brooding temperature the first week to induce heat stress, which significantly decreased broiler weight at day 7 but not at day 21 and 34. [Ferraz et al., 2017] investigated the effect of different durations of both heat and cold stress, and found that early heat stress at 36 °C did not negatively impact performance but cold stress at 27 and 30 °C did negatively impact performance. The temperature impact on weight gain

and feed per gain after week 3 was investigated in [May and Lott, 2001], where the effect of constant temperature of $\{12, 14, \ldots, 28, 30\}$°C until day 49 was investigated. Using regression formulas, it was found that high temperatures correlates with lower weight, and the optimal feed per gain depends on broiler weight, e.g., the best feed per gain was obtained using 27 °C for 0.85 kg broilers and 18 °C for 2.6 kg. [Blahová et al., 2007] found that subjecting 3 week old broilers to significantly colder temperatures, between 5-10 °C, did effect broiler weight at day 42 but had a significant negative impact on feed conversion ratio.

[Ferraz et al., 2018] used genetic fuzzy logic models to predict respiratory rate, an indicator of broiler health, using thermal stress duration and temperature. [Roushdy et al., 2018] studied the gene expression resulting from thermal stress in fast growing broilers (ROSS 308 and Cobb 500), with the intention of assisting artificial selection of more thermal stress resistant broilers strains.

2.1.2 Feed

Feed is the most expensive component of broiler production, and thus receives significant academic attention. See [Vieira and Angel, 2012] for a fairly recent review of the impact of different amino acid compositions on broiler performance. Similarly, see [Hippenstiel et al., 2011] for a review of the effect of herbs and essential oils, e.g. Thyme and Oregano, on growth performance. See [Neves et al., 2014] for a review of the biomechanical characteristics of broiler feeding, e.g., the impact of feed particle size.

Protein ingestion increases heat production compared to carbohydrate or fat[Musharaf and Latshaw, 1999]. Lowering metabolic heat production using temporary feed restriction[Lin et al., 2006] and dietary protein restriction[Zulkifli et al., 2018] has been proposed as an effective tool for improving heat resistance.

[Günal, 2012] studied the effect of short-term fasting, i.e. timed feed restriction, on prolonged heat exposure during the warmest part of the day on performance. Feed restriction decreased the broiler body temperature, but decreased growth performance on day 42 compared to ad libitum feeding – similar results was obtained in [Yalçin et al., 2001].

[Blahová et al., 2007] found that cold stressed broilers supplemented with more crude protein than required improved performance, i.e., increased weight and lower FCR. Replacing crude dietary protein with several essential amino acids under hot conditions has been found to improve performance in [Zaman et al., 2008] – crude protein was found to have a *quadratic* effect on weight and FCR. Under such hot and humid conditions, [Rahman et al., 2002] found that 21% protein may be suitable, as it produced the best performance at 8 weeks of age.

Hence, dietary composition is an important factor, and if given the choice from a range of feedstuffs, there is evidence for broilers being able to compose a diet matching their physiological need[Hughes, 1984]. [Syafwan et al., 2012] found that broilers given the choice between high protein diet, high energy diet

with low protein, and a control diet with medium protein and energy under heat stress had similar performance as only feeding the control diet.

Feeder height has not been reported to impact performance[Neves et al., 2014]. It has been proposed that too high feeder position decrease flock weight uniformity, as it limits feed access for smaller birds [Ferket and Gernat, 2006]. [Neves et al., 2015] studies behavior of broilers at different feeders using image analysis, and found a higher eating activity from feeders without partitioning grids. [Roll et al., 2010] found that broilers, if given the choice, prefer to consume feed from low height feeders, resulting in reduced eating frequency and time spent eating without affecting the total feed consumption.

2.1.3 Broiler Weight Measurements

Accurate broiler weight measurements are important for broiler management, as it assists the farmer in monitoring the broiler health, as well as scheduling broiler pick-up and subsequent slaughter. A negative weight bias between the automatic weighting systems and the manual weighting are quite common for broilers older than 15-21 days in industry. This problem was first reported by [Newberry et al., 1985], but the subject has subsequently received limited academic attention. [Chedad et al., 2000] hypothesized that the automatic weighting scale was used less frequently by heavier broilers through image analysis. This hypothesis was first confirmed in [Chedad et al., 2003] using an automatic image processing system, from which a statistically difference in the visible surface area was found between broilers visiting and broilers in the neighborhood of the weighting system. [Naas et al., 2009] studied lameness in broilers, i.e. abnormal gait or stance as a result of dysfunction of the locomotor system, and found that the walking ability of fast-growing broilers decreased with age, caused by unnatural biomechanical forces resulting in gait alteration. Similarly, [Tullo et al., 2017] found that sick, lame and very heavy birds reduce locomotor activity by extending the time spend in resting and lying positions. Hence, this helps explain why automatic weighing pads might fail at continuously tracking the broiler weight trend of the entire flock. Furthermore, it presents a potential challenge for the emergence of machine learning in broiler production, especially considering that broiler weight is one of the primary outputs. In practice, the biased broiler weight is corrected by multiplying a time dependent correction factor, typically up to 10%, which naturally introduces some uncertainties. Especially considering that the broiler weight, e.g., is positively correlated with parent age at day 7, 21 and 34 according to [Henriksen et al., 2016].

Alternatives to the automatic weighting pad system has been proposed in scientific literature. The image processing system in [Chedad et al., 2003] was used as a basis for estimating broiler weight in [Wet et al., 2003] through regression models of the surface area and broiler weight with a relative accuracy of 10% or 73g standard deviation – using broiler circumference resulted in a relative accuracy of 15% or 106g standard deviation. A similar result was

obtained with image processing in [Amraei et al., 2017] using an ellipse fitting algorithm and neural networks with lowest standard error of 82.4g. [Fontana et al., 2017] attempted to estimate broiler weight through sound analysis, and found that the peak frequency might be a predictor for broiler weight – however, it was only realistically considered to support weighting pads.

2.1.4 Pre-slaughter transport and catching

Pre-slaughter transport and catching causes great stress in broilers, and is know to have a large impact on animal welfare and meat quality – for a detailed review of poultry handling see [Weeks, 2014]. [de V. Queiroz et al., 2015] studied temperature increase as a function of stress, of which broilers where most susceptible to stress during and after catching. [dos Santos et al., 2017] studied the micro-climate, i.e. temperature and humidity, during pre-slaughter transportation, and found that meat quality was negatively influenced by long-distance transport during rainy season, and transport during dry season resulted in the best meat quality.

2.1.5 Lighting

Lighting sources and programs for broiler production is one of the most studied environmental variables in broiler production, and is widely utilized to regulate poultry production and welfare – perhaps because of its insensitivity to physical conditions. [Yang et al., 2018] recently conducted a meta study study on light intensity by extracting data from previous studies, and found that (<1 lux) resulted in loss of productivity, (<5 lux) lead to welfare concern but less feed consumption with same body weight as (>10 lux), and (>10 lux) lead to increased mortality and uniformity. [Yang et al., 2015] studied intermittent lighting, by comparing continuous lighting (24L:0D) with artificial day length of 2 (2L:2D) and 4 hours (4L:4D), where (xL:yD) indicates x hour light followed by y hours of dark. Performance in terms of FCR, average daily feed intake and body weight at day 42 was investigated, where continuous lighting produced the worst results with FCR of 2.38, (2L:2D) with FCR of 2.3 and (2L:2D) being the best with an FCR of 2.23. [Petek et al., 2005] studied the interaction between intermittent feeding and lighting schedules. Continuous lighting (24L:0D) was compared with natural light during the day and 3 cycles of (1L:3D) during night in ad Librium fed broilers – producing the sequence {(1L:3D), (1L:3D), (1L:3D), (12L:3D)}. Continuous ad libitum feeding (24F:0D) was compared to intermittent feeding (3F:3D), where (xF:yD) indicates x hours of ad libitum feeding followed by y hours of food deprivation. Intermittent lighting resulted in increased performance with a 113g higher weight and 0.16 lower FCR compared to continuous lighting, Intermittent feeding resulted in reduced performance with 58g lower weight and 0.08 higher FCR.

2.1.6 Parent Stock

Artificial selection of broilers has resulted in fast grower strains with a profound increase in growth performance in terms of growth rate and feed efficiency. Translating into both a 50% reduction in FCR and over 400% increase in broiler growth rate between 1957 and 2005 [Zuidhof et al., 2014], and a reduction in time to reach 1500g of live weight from 120 days in 1925 to 30 days in 2005[Bessei, 2006]. Popular examples of such strains include Cobb 500 and ROSS 308, which arguably are the most prevelent broilers on the marked today. For genetic selection review, see [Hunton, 2006] and [Zuidhof et al., 2014]. See [Tallentire et al., 2018] and [Bessei, 2006] for reviews on animal welfare concern.

Broiler weight has been found to be positively correlated with breeder age in [Sabry et al., 2016], [Ulmer-Franco et al., 2010], [Sabry et al., 2013], and [Henriksen et al., 2016]. [Ulmer-Franco et al., 2012] and [Ulmer-Franco et al., 2010] found that young breeders produce eggs with smaller yolks, which may be a disadvantage when reared under same conditions as eggs with bigger yolks from older breeders. [Sabry et al., 2013] found a significant interaction between breeder age and hatching time, with eggs of older breeders having longer hatching time.

2.1.7 Hatching Practice

Incubation of eggs under industrial conditions are carried out at a constant temperature of 37.8 °C and typically take 21 days. In nature, the temperature during incubation is known to fluctuate widely [Webb, 1987]. The effect of egg shell temperature during incubation has been the subject of much research, at least dating back to [Minne and Decuypere, 1984]. For a breakdown of the incubation process see [Decuypere et al., 2001].

[Ipek and Sozcu, 2016] studies the influence of three constant temperature-bands, i.e., low (33.3 to 36.7 °C), control (37.8 to 38.2 °C) and high (38.9 to 40.0 °C), during 10 to 18 days of incubation on welfare and performance. Both the low and high temperature group had a significant negative impact on both live weight and FCR on day 42, but not on day 0. Many health markers, like gait due to inhibited bone development, was also negatively impacted. A similar study was conducted in [Ipek et al., 2014].

[Piestun et al., 2008a] studied the effect of hot intermittent thermal manipulation during incubation to increase heat stress tolerance. Incubating eggs was subjected to 39.5 °C intermittently between day 7 and 16, i.e. during the development of the thyroid and adrenal axis – the control group was incubated at 37.8 °C. The intermittent group had significantly subsequent lower body temperatures post hatching between day 0 and 35, and was significantly more resistant to heat stress. Performance was not significantly impacted up to day 35, however, [Piestun et al., 2013] found a difference at day 70 using the same method. For a variation of intermittent thermal manipulation by the same research group, see [Piestun et al., 2008b], [Piestun et al., 2009] and [Piestun

et al., 2011]. Similar results are presented more recently in [Rakshit et al., 2016] and [Morita et al., 2016].

[Nyuiadzi et al., 2017] studies the effect of cold intermittent thermal manipulation during incubation to increase cold stress tolerance. With 36.6 °C 6 hours per day between day 10 and 18, 15 °C 30 minutes per day between day 18 and 19, and control temperature of 37.6 °C. Body temperature was significantly higher for the second group – similar results on cold thermodynamic tolerances are presented in [Piestun et al., 2008a].

[Maatjens et al., 2016b] studied the effect of applying incubation temperatures of $\{35.6, 36.7, 37.8, 38.9\}$ °C onwards of incubation day $\{15, 17, 19\}$. Broiler performance on day 7 in terms of FCR and body weight was significantly better for the 36.7 °C groups, which might be due to better intestinal development as postulated in [Maatjens et al., 2016a]. Note that earlier studies showed no difference in broiler performance early on, such as [Ipek et al., 2014] and [Ipek and Sozcu, 2016].

The effect of breeder age on broiler thermal sensitivity is less researched. However, [Hamidu et al., 2018] found that egg surface temperature is higher for older breeders, suggesting a subsequent interaction.

2.2 Broiler Modeling and Optimization

Broiler feed conversion rate (FCR) and growth optimization reduces the amount of feed, water and electricity required to produce a mature broiler. *State-of-the-art* broiler production optimization is throughly dominated by biological approaches, which will quickly become evident simply by comparing this section with Section 2.1.

Empirically motivated nonlinear growth curve models have traditionally been used to determine the evolution of broiler weight. These models have been extensively studied in scientific literature, such as [Aggrey, 2002], [Ahmadi and Mottaghitalab, 2007] and [Hasan Eleroğlu and Duman, 2014]. A growth curve in this context is a function of time with a fixed structure fitted to past broiler weight data. Common models include the Richerds Model and Gompertz-Laird Model that are described by 4 and 3 parameters respectively, where the parameters have biologically intuitive interpretations – such as time and size of maximum growth rate [Aggrey, 2002]. Such abstractions allow for easy evaluation and comparison of different treatments. See [Demuner et al., 2017] for common growth curve models and parameters for the most popular broiler strains. In [Lopes et al., 2008], the relationship between broiler house environment and production performance was investigated using a non-dynamic neural network. [Abreu et al., 2015] modeled the impact of heat stress and its duration on performance using fuzzy modeling. By perturbing the temperature of the second week from 30 °C to $\{24, 27, 33\}$ °C for up to 4 days, the FCR was accurately predicted. Likewise, in [Diez-Olivan et al., 2018] quantile regression forests-based modeling, a non-dynamic method, was among others used to predict broiler weight on week 3, 5 and 6 using broiler house

environment. However, it is not clear how to extract temporal performance, i.e. day-to-day growth prediction, from such non-dynamic models.

Dynamic broiler growth models are an extension of growth curves and have primarily been developed for control synthesis in scientific literature. In [Aerts et al., 2003] a time-variant online parameter estimation of a dynamic model was successfully applied to predict future weight up to 7 days ahead without a priori information. The model uses feed intake as input and weight as output and was used for model predictive control (MPC) in [Cangar et al., 2007]. This control algorithm was tested in a laboratory setting with a stocking density of 5.3 birds/m^2 and 20 birds/m^2, the latter to emulate farm scale density. The mean relative weight control error was 2.7% and 7.3% for the low and high-density experiments respectively – suggesting that farm scale broiler production is harder to both predict and control. A similar result was obtained in [Demmers et al., 2010], where a small differential recurrent neural network was used to model the feed quantity and control the broiler weight using nonlinear MPC. In [Stacey et al., 2004] a dynamic broiler weight model was developed and used information about feed uptake and composition of two feed types with known nutritional value. It was successfully used for broiler weight control and achieved results comparable to that of a stockman on farm scale broiler production with 30,000-40,000 broilers per house. [Cangar et al., 2006] among others used linear, non-linear recursive, and growth curve models to predict the slaughter weight up to 5 days prior to slaughter only using broiler weight measurements, and found that the linear model produced the best forecasts on two batches.

Hence, the available scientific literature on dynamic broiler models is *very sparse* and focuses exclusively on active broiler weight control by regulating feed uptake and composition, which traditionally favors simplistic models [Wathes et al., 2008; Aerts et al., 2003]. This is understandable, as feed is considered the biggest expense in broiler production. However, this research is incompatible with *state-of-the-art* broiler industry as it uses ad libitum feeding regimes, which specifically excludes feed uptake regulation. Regulation of environmental broiler house conditions, such as temperature, are compatible with industry, and if managed correctly are known to minimize the required feed, water, and electricity to produce a mature broiler. No prior scientific modeling literature has been found studying the complex dynamic interconnection between broiler weight and broiler house environmental conditions.

This could be caused by a risk averse broiler farmer mentality, which is understandable, as the broiler production is a business with proprietary production "secrets", caused by very slim profit margins. This results in a lack of publicly available datasets and scientific research of industrial-scale production, thus, mostly limiting research to lab-scale experiments such as [Cangar et al., 2007] – only [Aerts et al., 2003] and [Diez-Olivan et al., 2018] studies industrial-scale production. The lack of publicly available datasets with environmental conditions are particularly detrimental for research in dynamic broiler modeling with environmental conditions.

2.3 Dynamic Neural Networks

Due to the lack of mathematical broiler growth models, data driven system identification techniques offer a potential solution. The scope is limited to supervised nonlinear regression type neural networks in the following due to the target application. In this context, it is a nonlinear data driven modeling technique capable of approximating any non-discontinuous function or dynamic system, given sufficient complexity and data – for a thorough introduction see [Du and Swamy, 2014] and [Haykin, 1994].

2.3.1 Neural Networks

Neural networks are inspired by the biological brain, in that it is a graph of nodes, i.e. neurons, connected by weighted edges, i.e. synapses, and learns from data, i.e. experiences. The neurons are organized in multiple connected layers of neurons, and each neuron have an activation function. Typical structures include the multilayer perceptron model (MLP), radial basis functions and wavelets with the MLP model arguably being the most popular and widely used.

Learning, or training, is the processes of iteratively adjusting the network weights to minimizing a cost function, typically in the form of a sum of squares. Error Back Propagation is an efficient way of calculating the error gradient respect the network weights, and is a special case of automatic differentiation, which at least dates back to [Rumelhart et al., 1986]. Gradient-descent based algorithms often suffer from slow convergence, network stagnancy and a tendency of getting stuck in local minimia[Rehman and Nawi, 2011; Yu and Wilamowski, 2018]. Combining gradient descent with momentum is commonly used to overcome some of these challenges[Yu and Wilamowski, 2018]. This requires careful selection of the momentum gain, of which adaptive momentum gain algorithms exists, like [Rehman and Nawi, 2011]. Second order training algorithms have much faster convergence and use the Hessian matrix, i.e. curvature of the cost function, to perform better step size and direction estimates[Hagan and Menhaj, 1994; Yu and Wilamowski, 2018] – such as Newtons algorithm. Quasi-Newton methods dates back to [J. E. Dennis and Moré, 1977] and approximates the inverse Hessian using past gradient evaluations, where recent contributions combine it with stochastic gradient methods[Sohl-Dickstein et al., 2014] and Nesteroc's accelerated gradient [Ninomiya, 2017]. The Levenberg Marquardt algorithm dates back to [Marquardt, 1963] and combines the convergence speed of Newtons algorithm with the stability of Error Back Propagation[Yu and Wilamowski, 2018].

Gradient free training algorithms are generally based on stochastic and numerical approximation. The Weight Perturbation algorithm proposed by [Jabri and Flower, 1992] uses first order forward numerical differentiation, while the ALOPEX algorithm proposed by [Unnikrishnan and Venugopal, 1994] is a stochastic alternative that resembles simulated annealing. [Panagiotopoulos

et al., 2010] propose approximating the gradient in a random direction, intended for on-line training, and provides an overview of similar gradient free training algorithms. Other such alternatives include particle swarm optimization [Shen et al., 2006; Barmpalexis et al., 2018] initially proposed by [Kennedy and Eberhart, 1995] and genetic programming [Barmpalexis et al., 2018].

As the networks grow the need for scalable distributed optimization alternatives has gained popularity, where recent contributions based on alternating direction method of multipliers (ADMM) includes [Taylor et al., 2016], [Wang et al., 2018] and [Liang et al., 2018].

Cross validation is used to prevent the network from overfitting to the training data by validating its performance on separate test data[Haykin, 1994] – for a comprehensive review see [Arlot and Celisse, 2010]. The holdout method is the simplest cross validation method and performs a single split of training and testing data[Devroye and Wagner, 1979]. The leave-p-out method performs a separate training using an exhaustive split routine with every possible subset containing p testing samples – non-exhaustive variants also exists. Early stopping is a popular regularization technique that selects the network weights producing the lowest cost function error with the test data [Haykin, 1994]. Tikhonov, or weight, regularization prevent the network weights from growing arbitrarily large by introducing the weighted norm of the network weights in the cost function[Peng et al., 2015]. Bayesian regulation is a popular method for calculating the regularization gain[Burden and Winkler, 2008]. See [Pérez-Sánchez et al., 2016] for an overview of pruning, i.e. gradually removing network weights, and growing, i.e. gradually adding network weights. The optimal brain surgeon algorithm, a first order technique, is a popular pruning algorithm proposed by [Hassibi et al., 1993] – a second order variant using the inverse Hessian was propose by [Hassibi and Stork, 1993]. [Henríquez and Ruz, 2018] recently proposed a non-iterative punning algorithm based on Garson's algorithm.

Many different activation functions have been proposed for neural networks in scientific literature, which mostly is a trade-off between computational complexity, convergence rate and precision. See [Laudani et al., 2015] for a review of commonly used activation functions for feed forward neural network, including faster computation strategies using look up tables, piecewise linear approximations and weight transformation. See [Piekniewski and Rybicki, 2004] for a visual comparison of performance for common activation functions in MLP networks.

Proper data preprocessing can reduce the required model complexity, i.e. number of model parameters, which are expected to be more robust but may not necessarily have better predictive abilities [de Noord, 1994]. For an overview of different data processing techniques, see [Famili et al., 1997] or more recently [Gibert et al., 2016]. [Nawi et al., 2013] studied the effect of data pre-processing, limited to Min-Max, Z-Score and Scaling Normalization, on common variations of the back-propagation algorithm on benchmark classification datasets, of which the classification accuracy was found to both depend on the dataset and algorithm. Similarly, [Kuźniar and Zając, 2015] compares

different principal component analysis and data scaling, including Min-Max, with a non-dynamic neural network.

Dynamic neural networks, and machine learning in general for that matter, are highly application dependent and require expert process knowledge to reduce and restrict the complexity of the training process. Despite the versatility of neural networks, they tend to organize in terms of applications, as in modeling literature, rather than in techniques, as in control literature. Recent comprehensive application reviews include: drying processes[Aghbashlo et al., 2015], performance, salubrity, and security of cities, buildings, and infrastructures [Aguirre and Letellier, 2018b], medical image analysis[Anwar et al., 2018], pile driving[Fatehnia and Amirinia, 2018], winemaking[Moldes et al., 2015] agriculture[Liakos et al., 2018] and crop yield[Chlingaryan et al., 2018]. Results of using different neural networks topologies to a specific modeling problem are mainly presented, such as non-dynamic neural networks, dynamic neural networks and genetic algorithms. For instance, some drying applications include freeze-drying using MLP models and back-propagation[Drăgoi et al., 2013], modeling moisture content and temperature of thin clay slaps[Sander et al., 2003], and moisture content prediction of cassava crackers[Lertworasirikul and Tipsuwan, 2008].

2.3.2 Dynamic Neural Networks

Dynamic, also called recurrent, neural networks are a special class of neural networks which contain "hidden" temporal states and are useful for time series modeling. For a comprehensive overview of dynamic neural networks structures, see [Tsoi and Back, 1997] and [Tsoi, 1998]. Similarly, for a good overview of nonlinear model parameter representations and estimation techniques, see [Aguirre and Letellier, 2018a].

Dynamic neural networks have been successfully applied to model complex biological processes. Recent applications include algae growth prediction in a laboratory setting [Wang et al., 2015], prediction of bioethanol production in a bioreactor [Grahovac et al., 2016], bioreactor prediction [Nair et al., 2016], yeast fermentation modeling in a bioreactor [Nasimi and Irani, 2014], state estimation in a continuous bio reactor [Hernandez et al., 2013], and prediction of dissolved oxygen in wastewater treatment [Sadeghassadi et al., 2018].

2.4 Input Variable Selection

State-of-the-art industrial broiler production typically executes 5-8 batches per house per year, of which identical broiler houses in terms of physical construction, management and climate nonetheless consistently exhibit different growth and feed uptake performance. Hence, data from multiple houses cannot easily be pooled together, which limits modeling to a single house. A parameter drift caused by naturally changing production conditions, such as the broiler house deteriorating and the broiler and feed performance increases, prevents

indefinite data accumulation and renders old production data invalid. Hence, broiler production data is a very limited resource. Note that data quantity requirements increase exponentially with the number of inputs, input lags and output lags for the DNN [May et al., 2011]. To make best possible use of the limited production data, IVS is used to select as few significant inputs and lags as possible. No prior scientific literature on IVS of broiler growth and environmental variables has been found.

IVS is a critical step in data driven modeling, where appropriate input variables are selected from available data. The IVS discipline is diverse and has many perspectives and applications – for an overview of time series IVS see [May et al., 2011], and for feature selection see [Brown et al., 2012]. The scope is narrowed to model free filter type IVS algorithms given the target application – also known as filter type IVS. One class of statistical IVS is based on MI. By definition, mutual information $I(X;Y)$ is the reduction in uncertainty with respect to the random variable Y due to the observation of the random variable X [May et al., 2008b]. MI is often calculated by estimating the underlying marginal and joint probability density functions (PDFs) through kernel density estimation (KDE). Such an IVS algorithm was formulated in [Sharma, 2000], and modified in [May et al., 2008b] among others. It uses successive regression to remove selected information from the input candidates and target output and estimates partial mutual information (PMI), while relying on computationally heavy bootstrapping to determine a stopping criteria. Alternatively, the Copula entropy can be used to estimate MI[Chen et al., 2014].

Mutual information based IVS has been applied extensively to environmental modeling, such as IVS for prediction of rainfall [Sharma, 2000], salinity [Bowden et al., 2005], water quality [May et al., 2008a], storm water runoff [He et al., 2011], flood forecasting [Chen et al., 2014], and rainfall-runoff [Li et al., 2015]. Other recent applications include return temperature estimation in mixing loops [Overgaard et al., 2017], and Coriolis flow-meters for two-phase flow [Wang et al., 2017].

2.5 Iterative Learning Control

Broiler production is a biological batch process, which is known to be highly nonlinear and time varying, making traditional linear modeling and control tools inadequate [Bonné et al., 2004]. Iterative Learning Control (ILC) is a high precision control tool that iteratively improves the performance of repetitively operating dynamic systems, i.e systems executing a finite duration operation over and over again, which makes it a potential candidate for broiler growth optimization. It is a relatively recent but well-established field of study in control theory and can be traced back to [Uchiyama, 1978] and [Arimoto et al., 1984].

The objective for the batch process is to track a known reference trajectory as closely as possible. ILC solves this problem by applying feedback from the error of the current and past batches [Wang et al., 2009]. Once an execution

is completed, the system resets to the starting location and the next execution can begin. In the literature, each execution is termed a trial (or a pass or iteration) and the duration of each trial is known as the trial length. Let $u_k[n]$ be the control input, $r[n]$ the reference, $y_k[n]$ the output and $e_k[n]$ the tracking error – all of appropriate dimensions for trial k and sample n. Then the goal of ILC is to force the tracking error $e_k[n]$ to converge to 0 in k such that $y_k[n] = r[n]$ for $k \to \infty$ with a suitably designed control input $u_k[n]$. Hence the control action is chosen to sequentially improve performance from trial-to-trial. A simple structure ILC law was the basis of the results reported in [Arimoto et al., 1984] (using so-called D-type ILC) and many designs using such control laws have been experimentally validated. A starting point for the literature is the survey papers [Bristow et al., 2006] and [Ahn et al., 2007].

In [Duran-Villalobos and Lennox, 2013] ILC has been applied successfully to a fermentation process. In [Maeda et al., 2015] an ILC algorithm is combined with a disturbance observer to remove near-repetitive disturbances, and it is tested and verified on an excavator. Robust and Fault-tolerant ILC for a class of uncertain multiple input multiple output (MIMO) systems with actuator faults is presented in [Ding et al., 2015]. Other recent and noteworthy contributions include [Márquez-Vera et al., 2014], [Lim and Barton, 2014], and [Shen and Wang, 2014].

Model based ILC will be required in many cases and this area has been the subject of much research for both linear and nonlinear dynamic models. Norm optimal ILC is a popular branch of model based ILC, where the ILC input is optimal respect a cost function. The idea was formalized in [Furuta and Yamakita, 1986] and further developed in [Amann et al., 1996]. Noteworthy application examples include [Zhang, 2008], [van Zundert et al., 2016] and [Bolder and Oomen, 2015]. These include algorithms based on minimizing a suitably chosen cost function with applications including gantry robots, e.g., [Paszke et al., 2013], additive manufacturing, e.g., [Lim et al., 2017] and an extension to robotic-assisted stroke rehabilitation for the upper-limb with supporting clinical trials [Freeman et al., 2015].

Combining ILC schemes with a data driven modeling method is not a novel idea – recent examples include [Bolder and Oomen, 2015] and [Xu et al., 2013]. A similar combination of optimal ILC and ensemble dynamic neural network model, as used in this work, has been proposed in [Zhang, 2008].

Despite the versatility and robustness of ILC, its applications in scientific literature are mostly limited to repeating mechanical systems. It has not been applied to food production, let alone broiler production, of which the closest related ILC application arguably is bioreactors, such as [Márquez-Vera et al., 2014].

2.6 Contributions

The application of control theory techniques to broiler production has received very limited academic attention, of which available research is not applicable to *state-of-the-art* industrial broiler production. The main contributions of this thesis are outlined in this section, of which a simplified dependency chart is provided on Figure 2.1. All papers are provided in full in Part II.

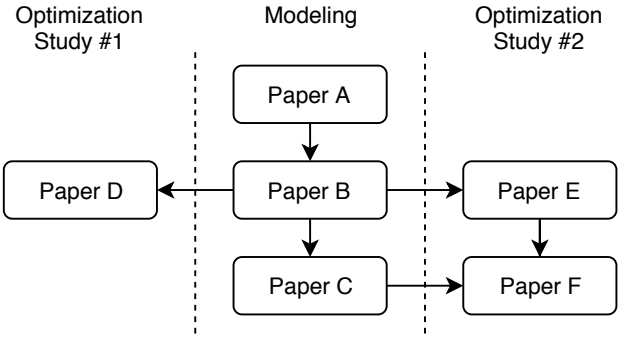


Figure 2.1: Simplified paper dependency chart grouped by research area.

2.6.1 Modeling

Paper A

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

Paper A is the first scientific literature investigating the dynamic influence of environmental broiler house conditions on broiler growth using dynamic neural network models. A pilot study is presented, where ensemble dynamic neural network forecasting models are trained on industrial farm scale production data from 12 batches from the same broiler house.

The model forecasts future broiler weight and uses environmental conditions such as heating, ventilation, and temperature along with broiler behavior such as feed and water consumption. Most notably, results indicate that the dynamic interconnection between environmental conditions and broiler growth adequately can be captured by the model.

The model is portrayed as a weight forecasting algorithm, which forecasts future broiler weights to key days of the production, as it could be a useful production management tool. Future inputs are necessary, of which comparable

forecasts are obtained using input data from the previous batch as a substitute for future input data.

Paper B

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Broiler weight forecasting using dynamic neural network models with input variable selection". *Journal of Computers and Electronics in Agriculture*, 2019c. doi:10.1016/j.compag.2018.12.014.

Paper B extends Paper A by including mutual information based input variable selection, which allows for automatic model structure configuration.

To make best possible use of the limited production data, IVS is used to select as few significant inputs and lags as possible for the dynamic neural network. To investigate the potential of the proposed method, an extensive case study on almost 3.5 years of industrial farm scale production data from a state-of-the-art broiler house is carried out. This effectively provides an overview of the most important environmental variables for dynamic broiler weight modeling.

The dynamic impact of environmental conditions on broiler growth is found to be significant and useful broiler weight forecasts are obtained – effectively providing a foundation for future research on optimization of broiler production.

Paper C

S. V. Johansen, J. D. Bendtsen, and J. Mogensen. "Broiler slaughter weight forecasting using dynamic neural network models". *Proceedings of the International Conference on Industrial Engineering and Applications*, 2019a. doi:10.1109/IEA.2019.8714850.

Paper C is concerned with both investigating and compensating for the negatively biased broiler weight measurements experienced onwards of day 15-21 mentioned in Section 2.1.3, which is a widely known and unsolved problem in *state-of-the-art* industrial broiler production. Paper C extends Paper A and Paper B by introducing and emphasizing the accurately measured broiler slaughter weight during training of the dynamic neural network model. The slaughter weight forecasting standard error is reduced from 162.4g to 65.4g on more than 4.5 years of *state-of-the-art* industrial production data. This paper furthermore provides insight into effective algorithm settings.

2.6.2 Optimization Study #1

Paper D

S. V. Johansen, J. D. Bendtsen, and J. Mogensen. "Broiler growth optimization using optimal iterative learning control". *Proceedings of the American Control Conference*, 2019b.

Paper D presents the first recorded attempt at optimizing FCR using norm optimal ILC, as it has been especially developed for systems that make repeated executions of the same finite duration task, in combination with the data driven model from Paper B. The proposed method regulates broiler weight and feed uptake using broiler house temperature under *state-of-the-art* production conditions.

Practical results of 4 trials obtained over 6 months of testing from a state-of-the-art broiler house with 40,000 broiler per trial is presented. A normalized maximum decrease in required feed of 2.5% is obtained over the two first trials, which is quite promising, but the subsequent 2 trials resulted in worse FCR, as the reference temperature could not be maintained. More trials are required.

2.6.3 Optimization Study #2

Paper E

S. V. Johansen, M. R. Jensen, B. Chu, J. D. Bendtsen, and E. Rogers. "Broiler growth optimization using norm optimal terminal iterative learning control". *Proceedings of Control Technology and Applications*, 2018.

Paper E presents a modified ILC algorithm that is more suitable for data driven broiler optimization based on point to point norm optimal ILC and the model from Paper B. It is formulated as a descend type algorithm to cope with the uncertain nature of the data driven model, and the reference is altered to automatically maximize the terminal broiler weight, i.e., slaughter weight.

To evaluate the proposed algorithm in simulation, especially considering the 6 months testing time required in Paper D, a heuristic broiler growth model based on the knowledge of a broiler application expert is formalized. An extensive simulation study shows that the proposed broiler weight algorithm maximizes the terminal broiler weight.

Paper F

S. V. Johansen, M. R. Jensen, B. Chu, J. D. Bendtsen, J. Mogensen, and E. Rogers. "Broiler FCR optimization using norm optimal terminal iterative learning control". *IEEE Transactions on Control Systems Technology*, 2019d. In review.

Paper F extends Paper E and Paper C. The terminal FCR is minimized compared to maximizing terminal weight in Paper E, of which feed uptake is added to the heuristic broiler model. The model from Paper C is used, of which a crude weight bias model is added to the heuristic broiler model.

An extensive simulation study shows that the proposed broiler weight algorithm can minimize terminal FCR, although, the weight bias is found to significantly impair convergence despite the compensation proposed in Paper C. Through simulation, it is demonstrated that iterative searching strategies, e.g. ILC, are required for broiler FCR minimization. The proposed method reduced feed consumption by 1.4% (non-normalized) on the same locations as Paper E.

A combined total improvement of 3.8% is obtained for both Paper D and Paper F – corresponding to a yearly saving of 8.500€ and 26.7 tones of broiler feed per broiler house.

In the following two chapters, a summary of the contributions of this study is presented.

3 Data Driven Broiler Modeling

This chapter provides an overview of the data driven broiler modeling efforts detailed in Paper A, Paper B and Paper C.

3.1 Mutual Information-based Input Variable Selection

3.1.1 Information Theory

The information entropy of the random variable X is given by

$$H(X) = -\int_X p(x)\ln p(x)dx, \tag{3.1}$$

and is a measure of how much information is contained by X. Note that this measure extends to multivariate inputs as well, in which case it is a measure of the collective information of all variables. The mutual information $I(X;Y)$ is the reduction in uncertainty with respect to the random variable Y due to the observation of the random variable X, and is given by

$$I(X;Y) = -\int_Y \int_X p(x,y)\ln\frac{p(x,y)}{p(x)p(y)}dx\,dy. \tag{3.2}$$

Alternatively, it can be regarded as the amount of information X provides about Y, and vice versa. It can be shown to be a function of the linear correlation coefficient $R_{XY} \in [-1;1]$ in case of linear and Gaussian distributed inputs given by

$$I(X;Y) = -\frac{1}{2}\ln\left(1-R_{XY}^2\right). \tag{3.3}$$

However, mutual information is more robust due to its insensitivity to noise and data transformations, i.e. nonlinearity [May et al., 2011]. Through resubstitution (3.1) and (3.3) can be estimated by

$$\hat{H}(X) = -\frac{1}{n}\sum_{k=1}^{n}\ln\hat{f}_X(X[k]) \text{ and} \tag{3.4a}$$

$$\hat{I}(X;Y) = \frac{1}{n}\sum_{k=1}^{n}\ln\frac{\hat{f}_{X,Y}(X[k],Y[k])}{\hat{f}_X(X[k])\hat{f}_Y(Y[k])}, \tag{3.4b}$$

where $\hat{f}_X(\cdot)$ and $\hat{f}_Y(\cdot)$ are the estimated marginal probability density function (PDF) of X and Y, respectively, and $\hat{f}_{X,Y}(\cdot,\cdot)$ is the estimated joint PDF of X and Y [Beirlant et al., 1997]. Intuitively, the nominator of (3.4b) equals the denominator if X and Y are uncorrelated, as $\hat{f}_X(X[k])\hat{f}_Y(Y[k]) = \hat{f}_{X,Y}(X[k],Y[k])$, and consequently does not add any new information, as

$\ln 1 = 0$. With $X = \{X_1, \ldots, X_i\}$, $Y = \{Y_1, \ldots, Y_j\}$ and $x \in \mathbb{R}^i$ the notation for multivariate mutual information and i-variate joint density estimation is

$$I(X;Y) \triangleq I(X_1, \ldots, X_i; Y_1, \ldots, Y_j) \text{ and}$$
$$\hat{f}_X(x) \triangleq \hat{f}_{X_1,\ldots,X_i}(x_1, \ldots, x_i).$$

The partial mutual information (PMI) is a measure of the information between two variables X and Y, excluding the information from a third variable Z. It can be regarded as a measure of how much mutual information is not explained by Z, which is useful for determining if an input variable provides additional information about an output variable. In terms of mutual information and information entropy it is given by:

$$\begin{aligned} I(X;Y \mid Z) &= I(X;Y,Z) - I(X;Z) \\ &= H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) \end{aligned} \tag{3.5}$$

The relationship between information entropy, mutual information and partial mutual information is visualized on Figure 3.1.

3.1.2 Kernel Density Estimation

Kernel density estimation is employed to calculate the joint PDFs in (3.4). The d-variate Gaussian kernel density estimator (KDE) can be formulated by

$$\hat{f}_X(x) = \frac{1}{n} \sum_{k=1}^{n} K_{\bar{H}}(x - X[k]) \text{ with} \tag{3.6a}$$

$$K_{\bar{H}}(y) = \frac{K\left(\bar{H}^{-1} y\right)}{\det\{\bar{H}\}} \text{ and} \tag{3.6b}$$

$$K(z) = \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2} z^T z\right). \tag{3.6c}$$

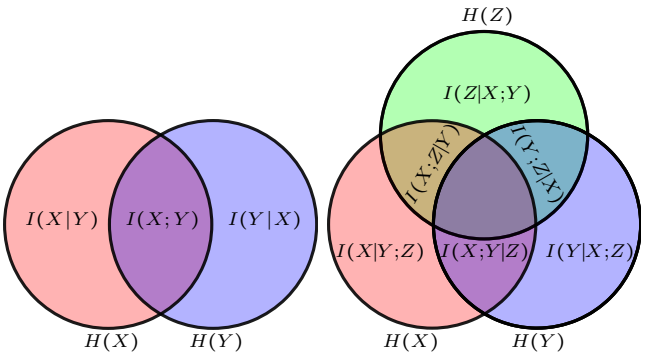
Where (3.6a) and (3.6b) are the multivariate parzen window with $x, y, z \in \mathbb{R}^d$, $\bar{H} \in \mathbb{R}^{d \times d}$ being the kernel bandwidth or smoothing matrix, n is the number of samples in the known data $X \in \mathbb{R}^{d \times n}$, and $X[k] \in \mathbb{R}^d$ denotes the k'th sample of X. In this work the generalized Scott's rule[Wolfgang Härdle and Werwatz, 2004, pp. 73] is used. It is given by

$$\bar{H} = \Sigma^{1/2} \bar{h}_S \text{ with } \bar{h}_S = n^{-1/(d+4)}, \tag{3.7}$$

where $\Sigma \in \mathbb{R}^{d \times d}$ is the covariance matrix of the known data $X \in \mathbb{R}^{d \times n}$ and $\bar{h}_S \in \mathbb{R}$. Furthermore, (3.6c) is known as the multivariate Gaussian kernel density function. The full expression for (3.6) equals

$$\hat{f}_X(x) = \sum_{k=1}^{n} \frac{\exp\left(-\frac{1}{2}(x - X[k])^T \left(\bar{h}_S^2 \Sigma\right)^{-1} (x - X[k])\right)}{n\sqrt{(2\pi)^d \det\left(\bar{h}_S^2 \Sigma\right)}}. \tag{3.8}$$

Applying it on a dataset is illustrated on Figure 3.2.



(a) MI **(b)** PMI

Figure 3.1: Venn diagram representation of MI and PMI.

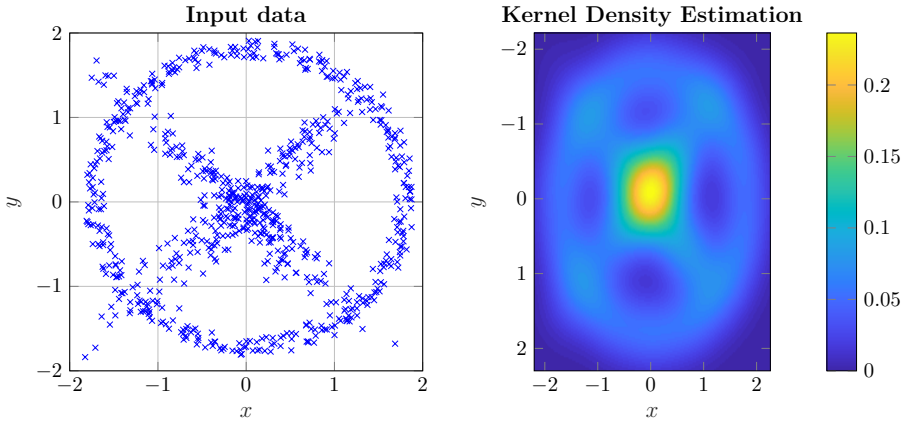


Figure 3.2: The estimated probability density distribution $\hat{f}_{X,Y}(x, y)$ given by (3.8) of two correlated variables x and y. The linear correlation coefficient is -0.005 in this case, suggesting that x and y are very close to being uncorrelated. The KDE represents the underlying correlated distribution significantly better in this case. The Mutual Information $\hat{I}(x; y) = 0.204$ given by (3.4b), suggesting that information of y can be obtained from x, as the 95% critical value for Gaussian noise with $n = 400$ samples equals 0.0567 [May et al., 2008a].

3.1.3 Input Variable Selection (IVS)

The task of progressively selecting input variables can be formulated as follows. Let X denote the set of lagged input candidates, $S = \{S_1, \ldots, S_{m-1}\}$ the $(m-1)$ previously selected lagged inputs with $S \subseteq X$ and Y being the target output. The m'th lagged input candidate with *most new information* is

selected according to

$$S_m = \underset{C \in X \setminus S}{\arg\max}\, I(C, S; Y) = \underset{C \in X \setminus S}{\arg\max}\, I(C; Y \mid S). \tag{3.9}$$

This is repeated until S_m provides *no significant* new information about Y, and S_m is not added to S. Originally, the 95 percentile critical bootstrap value was used to estimate a termination threshold, but since it is computationally heavy [May et al., 2008b] compared alternatives. Termination criterias based on the Akaike Information Criterion, modified Hampel outlier test and PMI lookup table based on Monte Carlo simulations of normally distributed noise for different sample sizes were compared. Satisfactory results are obtained using the 95-percentile tabulated PMI as depicted on Figure 3.3, but it should be noted that [May et al., 2008b] found that this method gives biased values for non-Gaussian distributed inputs. Realizing that this table is approximately log-log linear in the number of samples n, the following approximation is found (See Figure 3.3):

$$\hat{I}^{(95)}(n) = 1.7134 \cdot n^{-0.518} \tag{3.10}$$

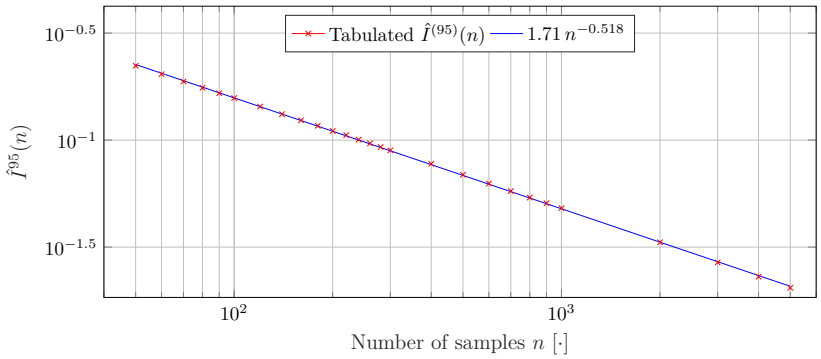


Figure 3.3: Tabulated PMI and an approximation [May et al., 2008b].

Calculation of (3.9) requires estimation of an $(m + 1)$ dimensional PDF using (3.8), and is therefore increasingly dependent on the data quality for larger m – both in terms of number of samples and increasing data sparsity in higher dimensional spaces. This phenomenon is called *the curse of dimensionality*. Hence, calculation of higher order MI is not preferred, as no data quality guarantees can be made for broiler production.

3.1.4 Regression based PMI

This method is taken from [May et al., 2008b] and uses the KDE based conditional expectation known as the Nadaraya-Watson estimator for regression.

It is the regression of Y given the observed X, denoted by $E[Y|X]$, where the j'th sample is calculated by

$$E[Y \mid X = X[j]] = \frac{1}{n} \frac{\sum_{k=1}^{n} Y[k] K_{\bar{H}}(X[j] - X[k])}{\sum_{k=1}^{n} K_{\bar{H}}(X[j] - X[k])}. \tag{3.11}$$

For simplicity, the removal of known "information" through regression of the elements of B from the variable A is written as

$$M(A|B) = A - E[A|B].$$

The key idea behind this method is to estimate the partial mutual information by removing the known mutual information from the selected inputs through successive regression. Let $(\cdot)^i$ denote the i'th successive regression of $(\cdot)$, then the m'th iteration of the selected inputs $S = \left\{S_1^1, \ldots, S_{m-1}^{m-1}\right\}$ of $X_k \in X$ equals

$$X_k^m = \begin{cases} M\left(X_k^{m-1} | S_{m-1}^{m-1}\right) & m > 1 \\ X_k & m = 1 \end{cases}. \tag{3.12}$$

Notice that X_k^m depends on up to m regressions – one for each previously selected element of S. To demonstrate, X_k^4 is obtained from the following successive regressions of the previously selected variables, $\{S_3, S_2, S_1\}$, according to:

$$\begin{array}{llll} X_k^1 = X_k & S_3^1 = S_3 & S_2^1 = S_2 & S_1^1 = S_1 \\ X_k^2 = M\left(X_k^1 | S_1^1\right) & S_3^2 = M\left(S_3^1 | S_1^1\right) & S_2^2 = M\left(S_2^1 | S_1^1\right) & \\ X_k^3 = M\left(X_k^2 | S_2^2\right) & S_3^3 = M\left(S_3^2 | S_2^2\right) & & \\ X_k^4 = M\left(X_k^3 | S_3^3\right) & & & \end{array}$$

Note that the first row contains the original values, and each subsequent row is a regression based only on the previous row.

Using (3.12), the IVS problem formulated in (3.9) can be approximated by

$$S_m = \underset{C \in X \setminus S}{\arg\max} \left[I(C; Y \mid S) \approx I\left(C^{m-1}; Y^{m-1}\right) \right], \tag{3.13}$$

which has the benefit of only requiring 2-dimensional density function estimates compared to the previous $(m-1)$. When implementing this algorithm, the values of $(\cdot)^m$ are calculated from $(\cdot)^{m-1}$ after S_m has been selected to avoid unnecessary recalculation.

3.2 Dynamic Neural Network

3.2.1 Model

The particular type of dynamic neural network (DNN) model used in this work can be classified as a discrete-time nonlinear ARMAX model given by

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = \mathcal{N}\big(\hat{Y}[k], U[k] \mid \mathcal{W}\big) \qquad \forall p \le k \tag{3.14a}$$

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = y[k+1 \mid b] \qquad \forall k < p \tag{3.14b}$$

with

$$U[k] = \begin{bmatrix} u[k-\bar{n}_1 \mid b]^T & \cdots & u[k-\bar{n}_{N_{\bar{n}}} \mid b]^T \end{bmatrix}^T$$

$$\hat{Y}[k] = \begin{bmatrix} \hat{y}[k-\bar{m}_1 \mid b,p,\mathcal{W}]^T & \cdots & \hat{y}[k-\bar{m}_{N_{\bar{m}}} \mid b,p,\mathcal{W}]^T \end{bmatrix}^T$$

where $\hat{y}[k \mid b,p,\mathcal{W}] \in \mathbb{R}^{N_y}$ is the model output at sample $k \in \mathbb{Z}_+$, initialized with data from batch $b \in \mathbb{Z}$ at sample $p \in \mathbb{Z}_+$ with the $N_{\mathcal{W}}$ model weights $\mathcal{W} \in \mathbb{R}^{N_{\mathcal{W}}}$, $y[k \mid b] \in \mathbb{R}^{N_y}$ is the measured output and $u[k \mid b] \in \mathbb{R}^{N_u}$ is the measured input at batch b and sample k, $\mathcal{N}: \mathbb{R}^{N_y N_{\bar{m}}} \times \mathbb{R}^{N_u N_{\bar{n}}} \to \mathbb{R}^{N_y}$ is a multilayer perceptron model (MLP) model, $U[k] \in \mathbb{R}^{N_u N_{\bar{n}}}$ is delayed values of the input vector $u[k \mid b]$ corresponding to the $N_{\bar{n}} \in \mathbb{Z}_+$ elements of $\bar{n} = \{\bar{n}_1, \ldots, \bar{n}_{N_{\bar{n}}}\}$, and $\hat{Y}[k] \in \mathbb{R}^{N_y N_{\bar{m}}}$ is delayed values of the past output vector $\hat{y}[k \mid b,p,\mathcal{W}]$ corresponding to the $N_{\bar{m}} \in \mathbb{Z}_+$ elements of $\bar{m} = \{\bar{m}_1, \ldots, \bar{m}_{N_{\bar{m}}}\}$. The sample number k and initialization sample p for the model output $\hat{y}[k \mid b,p,\mathcal{W}]$ is bounded by $p \in [1; N_{s,b}]$ and $k \in [p+1; N_{s,b}]$ where $N_{s,b} \in \mathbb{Z}_+$ is the number of samples in batch b. This particular structure has been adopted to accommodate potentially long propagation delays, while keeping the number of weights relatively low. Model initialization occurs through (3.14b), where k is implicitly lower bounded by 1 for both $y[k \mid b]$ and $u[k \mid b]$.

The DNN model $\mathcal{N}$ is selected with one hidden layer with hyperbolic tangent activation function in the hidden layer and linear activation function in the output layer. In matrix-vector representation, (3.14a) is written explicitly as

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = W^{\mathrm{o}} \tanh\big(\mathcal{X} + \theta^{\mathrm{h}}\big) + \theta^{\mathrm{o}} \tag{3.16}$$

with

$$\mathcal{X} = \sum_{i=1}^{N_{\bar{m}}} W^{\mathrm{h}}_{y,i} \hat{y}[k-\bar{m}_i \mid b,p,\mathcal{W}] + \sum_{j=1}^{N_{\bar{n}}} W^{\mathrm{h}}_{u,j} u[k-\bar{n}_j \mid b],$$

where $N_h \in \mathbb{Z}_+$ is the number of neurons in the hidden layer, $\mathcal{X} \in \mathbb{R}^{N_h}$, $W^{\mathrm{o}} \in \mathbb{R}^{N_y \times N_h}$ is the output weights, $W^{\mathrm{h}}_{y,i} \in \mathbb{R}^{N_h \times N_y}$ is the delayed output weights, $W^{\mathrm{h}}_{u,j} \in \mathbb{R}^{N_h \times N_u}$ is the delayed input weights, $\theta^{\mathrm{h}} \in \mathbb{R}^{N_h}$ is the hidden layer bias and $\theta^{\mathrm{o}} \in \mathbb{R}^{N_y}$ is the output bias. A visual representation of (3.16) is depicted on Figure 3.4.

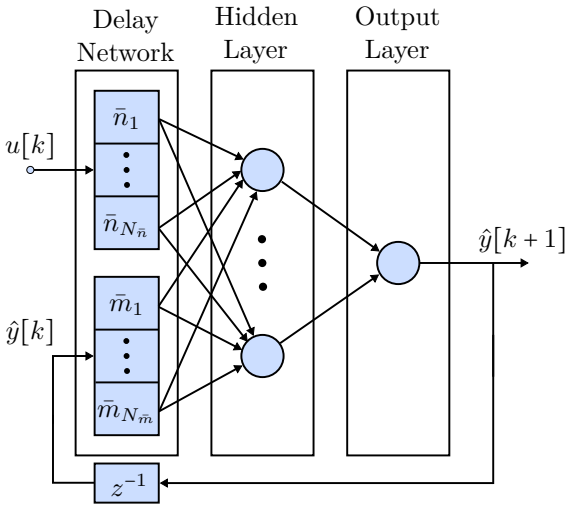


Figure 3.4: Visual representation of (3.16) with one input and one output for simplicity. Neurons are represented with blue circles, which contain an activation function and bias each. Note that all incoming signals to a neuron are multiplied by a weight. The operator z^{-1} produces a delay of 1 sample, while a blue box produce a delay equal to its number.

3.2.2 Input Variable Selection

Input variable selection is used to determine the structure of the DNN model based on Section 3.1. In this context, let $\tilde{u}$ and $\tilde{y}$ denote the set of potential inputs and outputs for the batch indexes $\mathcal{B}$. Similarly, let $N_{\tilde{u}}(\mathcal{B})$ and $N_{\tilde{y}}(\mathcal{B})$ denote the number of inputs and outputs of the batch indexes $\mathcal{B}$.

The input candidates X are constructed from both the potential input $\tilde{u} \in \mathbb{R}^{N_{\tilde{u}}(\mathcal{B})}$ and output $\tilde{y} \in \mathbb{R}^{N_{\tilde{y}}(\mathcal{B})}$ candidates from $\mathcal{B}$ according to

$$\begin{aligned} X = \big\{ \tilde{u}_i\big[k - \tilde{k}\big] \mid \tilde{k} \in [1; N_{L,\tilde{u}}] \wedge i \in [1; N_{\tilde{u}}(\mathcal{B})] \big\} \cup \cdots \\ \big\{ \tilde{y}_i\big[k - \tilde{k}\big] \mid \tilde{k} \in [1; N_{L,\tilde{y}}] \wedge i \in [1; N_{\tilde{y}}(\mathcal{B})] \big\} \end{aligned} \tag{3.17}$$

where $\tilde{u}_i[k - \tilde{k}]$ or $\tilde{y}_i[k - \tilde{k}]$ denotes the i'th element of $\tilde{u}$ or $\tilde{y}$ delayed by $\tilde{k} \in \mathbb{Z}_+$ samples, and $N_{L,\tilde{y}} \in \mathbb{Z}_+$ and $N_{L,\tilde{u}} \in \mathbb{Z}_+$ denotes the maximum output and input lag, respectively. This results in a total of $N_{L,\tilde{u}} N_{\tilde{u}}(\mathcal{B}) + N_{L,\tilde{y}} N_{\tilde{y}}(\mathcal{B})$ candidates. When calculating the KDE from X using (3.8), data from all batches are used simultaneously. To account for the largest lag, equivalent of $N_{L,\max} = \max\{N_{L,\tilde{u}}, N_{L,\tilde{y}}\}$, the first $N_{L,\max}$ samples are excluded from each batch – resulting in a total of $\sum_{b \in \mathcal{B}} (N_{s,b} - N_{L,\max})$ samples.

A separate IVS for each $\tilde{y}_i \in \tilde{y}$ is carried out, where IVS with $Y = \tilde{y}_i$ of the potential candidates X from (3.17) is denoted ${}^{i}S$. The overall IVS results in

$$\mathcal{S} = \bigcup_{i=1}^{N_{\tilde{y}}(\mathcal{B})} {}^{i}S, \tag{3.18}$$

which is used to configure the DNN model given by (3.14a) as follows. The

resulting selected outputs and inputs used by the DNN model are given by

$$u = \{\bar{u} \in \tilde{u} \mid k \in [0; N_{L,\tilde{u}}] \wedge \bar{u}[n-k] \in \mathcal{S}\} \text{ and} \tag{3.19a}$$

$$y = \tilde{y}. \tag{3.19b}$$

The input and output lags in (3.16) are given by

$$\bar{n} = \left\{\tilde{k} \in [0; N_{L,\tilde{u}}] \mid i \in [1; N_{\tilde{u}}(\mathcal{B})] \wedge u_i[k-\tilde{k}] \in \mathcal{S}\right\} \tag{3.20a}$$

$$\bar{m} = \left\{\tilde{k} \in [0; N_{L,\tilde{y}}] \mid i \in [1; N_{\tilde{y}}(\mathcal{B})] \wedge y_i[k-\tilde{k}] \in \mathcal{S}\right\} \tag{3.20b}$$

The weights $W^{\mathrm{h}}_{u,\tilde{k}}$ and $W^{\mathrm{h}}_{y,\tilde{k}}$ from (3.16) is configured as follows. Column $i \in [1; N_u]$ of $W^{\mathrm{h}}_{u,\tilde{k}}$ with delay $\tilde{k}$ is non-zero if $u_i[k-\tilde{k}] \in \mathcal{S}$, and column $i \in [1; N_y]$ of $W^{\mathrm{h}}_{y,\tilde{k}}$ with delay $\tilde{k}$ is non-zero if $y_i[k-\tilde{k}] \in \mathcal{S}$. The weights W^{o}, θ^{o} and θ^{h} are always fully populated. For example, if the inputs indexed 1 and 3 are selected with delay of $j = 2$, $N_u = 4$ inputs, $N_h = 3$ hidden neurons, then $W^{\mathrm{h}}_{u,j}$ equals

$$W^{\mathrm{h}}_{u,2} = \begin{bmatrix} \mathcal{W}_1 & 0 & \mathcal{W}_2 & 0 \\ \mathcal{W}_3 & 0 & \mathcal{W}_4 & 0 \\ \mathcal{W}_5 & 0 & \mathcal{W}_6 & 0 \end{bmatrix}. \tag{3.21}$$

3.2.3 Model Training

The model weights are found through training by solving

$$\mathcal{W}(\mathcal{B}, c) = \arg\min_{\tilde{\mathcal{W}}} \sum_{b \in \mathcal{B}\backslash c} \frac{E_b(\tilde{\mathcal{W}})}{\#\mathcal{B} - 1} \tag{3.22}$$

with

$$E_b(\tilde{\mathcal{W}}) = \sum_{p=1}^{N_{\mathcal{P}}} \sum_{k=\mathcal{P}_p}^{N_{s,b}} \frac{\|y[k \mid b] - \hat{y}[k \mid \mathcal{P}_p, b, \tilde{\mathcal{W}}]\|_2^2}{N_{r,b}} + \bar{\alpha}\|\tilde{\mathcal{W}}\|^2$$

$$N_{r,b} = \sum_{p=1}^{N_{\mathcal{P}}} \sum_{k=\mathcal{P}_p}^{N_{s,b}} N_y = N_{\mathcal{P}} N_y (N_{s,b} + 1) - N_y \sum_{p=1}^{N_{\mathcal{P}}} \mathcal{P}_p$$

where $\mathcal{W}(\mathcal{B}, c) \in \mathbb{R}^{N_{\mathcal{W}}}$ is the trained weights with the $\#\mathcal{B} - 1$ batch indexes in $\mathcal{B}\backslash c$, batch $c \in \mathcal{B}$ is used for cross validation, $\tilde{\mathcal{W}} \in \mathbb{R}^{N_{\mathcal{W}}}$ is the potential weights, $N_{r,b} \in \mathbb{Z}_+$ is the total number of residuals in $E_b \in \mathbb{R}_+$ with batch index b, and $N_{s,b} \in \mathbb{Z}_+$ is the number of samples in batch b. $\|\cdot\|_2$ is the standard Euclidean 2-norm. (3.22) is solved using the Levenberg-Marquardt optimization algorithm by means of the Ceres Solver library [Sameer Agarwal and Others, 2015] – see Paper B for details. Both the inputs and outputs are normalized to a mean of 0 and standard deviation of 1 during training. The network weights $\tilde{\mathcal{W}}$ are initialized using the Nguyen-Widrow algorithm as explained in [Nguyen and Widrow, 1990].

Each batch is divided into $N_{\mathcal{P}} \in \mathbb{Z}_+$ Sub-Batches with initial starting times denoted by $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_{N_{\mathcal{P}}}\}$, which tend to speed up training and decreases the risk of converging to a poor local minima. Consequently, for each batch, $b \in \mathcal{B}$, $N_{\mathcal{P}}$ sets of trajectories are generated with different initial conditions to minimize, as illustrated on Figure 3.5, where each output sample is weighted equally.

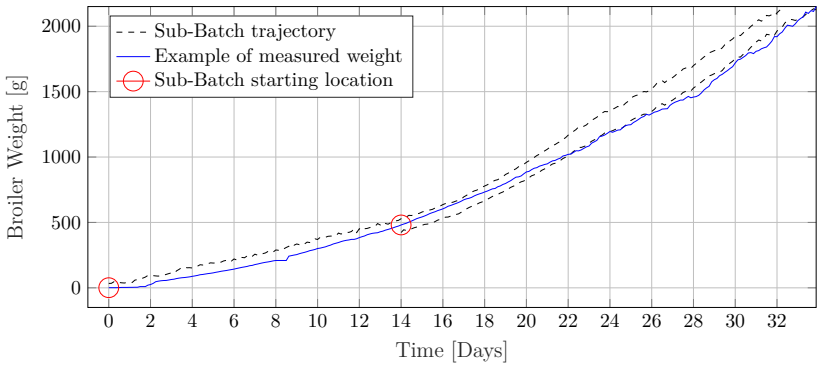


Figure 3.5: Example of the Sub-Batch trajectories with Sub-Batch starting time at day $\mathcal{P} = \{0, 14\}$.

The last term of the cost function is a scalar regularization term punishing the size of the $N_{\mathcal{W}}$ system weights $\mathcal{W}$, where $\bar{\alpha} \in \mathbb{R}_+$ is the regularization weight. The regularization weight $\bar{\alpha}$ is determined iteratively through Bayesian Regulation as described in [Burden and Winkler, 2008] – see Paper B for details.

Cross validation in combination with early stopping is applied to avoid overfitting and increase the models ability to generalize to inputs not present in the training data. This is facilitated by selecting the model weights $\tilde{\mathcal{W}}$ among all training iterations with the smallest cost $E_c(\tilde{\mathcal{W}})$ calculated with batch $c \in \mathcal{B}$. Note that this batch is only used for early stopping and not for training of $\tilde{\mathcal{W}}$. This type of cross validation could accurately be termed leave-one-batch-out cross validation, but is a special case of the holdout cross validation method – see [Arlot and Celisse, 2010] for a comprehensive review of cross validation methods.

3.3 Broiler Weight Forecasting

This section describes how the DNN model is used for broiler weight forecasting along with how the forecasts are evaluated.

Forecasting Trial Setup

The aim of forecasting trial $t \in \{1, \ldots, N_t\}$ is to forecast future outputs $p < k < N_{s,t}$ from the present moment $p \in \mathbb{Z}_+$ throughout trial t, only based on

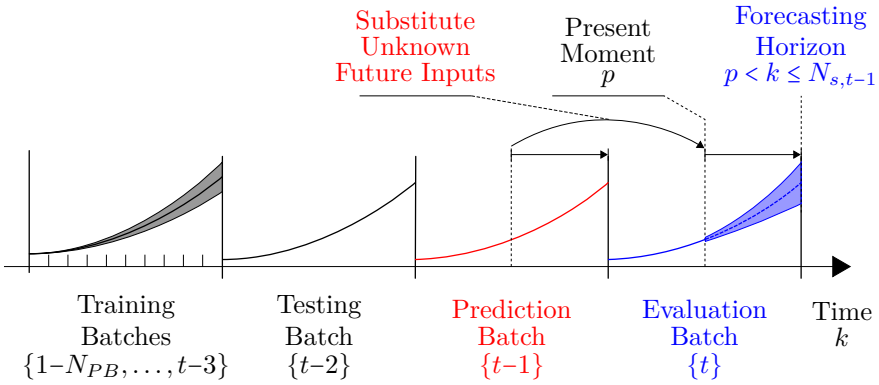


Figure 3.6: Visual representation of how batches are selected for forecasting trial t. The training and testing batches $\{N_{PB}, \ldots, t-2\}$ are used to train a model as described in Section 3.2. When forecasting future outputs at the evaluation batch $\{t\}$ ahead of the present moment $p < k$, then substitute future inputs from the prediction batch $\{t-1\}$ are used up to sample $N_{s,t-1}$ in the evaluation batch $\{t\}$.

past batches $\{1 - N_{PB}, \ldots, t-1\}$ with N_{BP} preliminary batches prior to trial $t = 1$. The batches are segmented into the following categories, as depicted on Figure 3.6, and equals: evaluation batch $\{t\}$, prediction batch $\{t-1\}$, testing batche $\{t-2\}$, and the remaining batches are denoted training batches $\{1 - N_{PB}, \ldots, t-3\}$. The training and testing batches are used to find the network weights $\mathcal{W}$ through training. The latest training batch is used in an early stopping setting to avoid over-fitting. In order to forecast future outputs on the evaluation batch $\{t\}$, unknown future input values are required, of which the prediction batch $\{t-1\}$ inputs are used as substitute for sample $k \leq N_{s,t-1}$.

3.3.1 Evaluation and Prediction Forecasting

When forecasting future outputs from sample p to sample k at trial t, then future inputs $u[k \mid t]$ are required due to the dynamic model structure. To alleviate this, the known "future" inputs from the past batch $t-1$ is used as substitute for the unknown future inputs for batch t by modifying (3.14) to

$$\hat{y}^*[k+1 \mid t, p, \mathcal{W}] = \begin{cases} \mathcal{N}\big(\hat{Y}^*[k], U^*[k] \mid \mathcal{W}\big), & p \leq k \\ y[k+1 \mid t], & k < p \end{cases} \tag{3.23a}$$

$$u^*[k \mid t] = \begin{cases} u[k \mid t-1] & p < k \\ u[k \mid t] & k \leq p \end{cases} \tag{3.23b}$$

with

$$U^*[k] = \begin{bmatrix} u^*[k - \bar{n}_1 \mid t]^T & \cdots & u^*[k - \bar{n}_{N_{\bar{n}}} \mid t]^T \end{bmatrix}^T \text{ and}$$

$$\hat{Y}^*[k] = \begin{bmatrix} \hat{y}^*[k - \bar{m}_1 \mid t, p, \mathcal{W}]^T & \cdots & \hat{y}^*[k - \bar{m}_{N_{\bar{m}}} \mid t, p, \mathcal{W}]^T \end{bmatrix}^T$$

where $\hat{y}^*[k+1 \mid t, p, \mathcal{W}]$ and $u^*[k \mid t]$ is the prediction forecasting output and input. When forecasting using (3.23), i.e. $\hat{y}^*[k+1 \mid t, p, \mathcal{W}]$, it is denoted *prediction forecasting*, and when using (3.14), i.e. $\hat{y}[k+1 \mid t, p, \mathcal{W}]$, it is denoted *evaluation forecasting*. The key difference between (3.14) and (3.23) is the use of $u[k \mid t-1]$ in (3.23b), as it ensures that the unavailable $u[k \mid t]$ for $p < k$ is not used for forecasting – this is also explained on Table 3.1. Note that only (3.14) is used for training, and that both evaluation and prediction forecasts use the exact same model weights $\mathcal{W}$.

	Past ($k \leq p$)	Future ($p < k$)
U	t	t
Y	t	$\approx t$

(a) Evaluation

	Past ($k \leq p$)	Future ($p < k$)
U	t	$t-1$
Y	t	$\approx t$

(b) Prediction

Table 3.1: The objective is to forecast future outputs Y for $p < k$ at trial t – denoted by $\approx t$. Evaluation forecasting (a) unrealistically relies on future inputs U for $p < k$ at trial t, while prediction forecasting (b) realistically relies on past inputs U for $p < k$ at trial $t-1$. The difference between the two methods has been highlighted with a gray box. Note that both forecasting approaches rely on past inputs and outputs for initialization, i.e., U and Y for $k \leq p$.

As all inputs and outputs are not guaranteed to be present in all the batches for trial t, up to $N_B \in \mathbb{Z}_+$ potential batches are selected for the IVS algorithm by maximizing

$$\begin{aligned} \mathcal{B}_t = \arg\max_{\tilde{\mathcal{B}}} \; & N_{\tilde{u}}(\tilde{\mathcal{B}})\, N_{\tilde{y}}\big(\tilde{\mathcal{B}}\big)\, \min\{\#\tilde{B}, N_B\} + \frac{1}{\max\{\tilde{\mathcal{B}}\} - \min\{\tilde{\mathcal{B}}\} + 1} \\ \text{s.t.} \quad & \tilde{\mathcal{B}} \subseteq \{1 - N_{PB}, \ldots, t-2\} \end{aligned} \tag{3.25}$$

where $\mathcal{B}_t$ is the set of batches used for IVS and training of trial t, $\tilde{\mathcal{B}}$ is a set of potential batch indexes, $N_{\tilde{u}}(\tilde{\mathcal{B}})$ and $N_{\tilde{y}}(\tilde{\mathcal{B}})$ are the number of potential inputs and outputs with batch indexes $\tilde{\mathcal{B}}$. As the broiler house naturally changes over time, a forgetting factor is introduced by upper bounding the number of potential training batches to N_B through the term $\min\{\#\tilde{B}, N_B\}$. The term $\frac{1}{\max\{\tilde{\mathcal{B}}\} - \min\{\tilde{\mathcal{B}}\} + 1} < 1$ ensures that the most recent batches take priority if more than N_B potential batches can be selected, where $\min\{\tilde{\mathcal{B}}\}$ is the smallest and $\max\{\tilde{\mathcal{B}}\}$ is the biggest index in $\tilde{\mathcal{B}}$.

Ensemble forecasts are generated with N_m sub-models trained with different initial weights on the same training data denoted $\{\mathcal{W}_1(\mathcal{B}_t, t-2), \ldots, \mathcal{W}_{N_m}(\mathcal{B}_t, t-2)\}$, which requires batch $t-2$ to be selected. The ensemble evaluation and prediction forecasting mean of output index $l \in \mathbb{Z}_+$ are respectively

given by

$$\hat{\bar{y}}_l[k \mid t,p] = \frac{1}{N_m}\sum_{i=1}^{N_m} \hat{y}_l[k \mid t,p,\mathcal{W}_i(\mathcal{B}_t,t-2)] \text{ and} \tag{3.26a}$$

$$\hat{\bar{y}}^*_l[k \mid t,p] = \frac{1}{N_m}\sum_{i=1}^{N_m} \hat{y}^*_l[k \mid t,p,\mathcal{W}_i(\mathcal{B}_t,t-2)], \tag{3.26b}$$

where $\mathcal{W}_i(\mathcal{B}_t,t-2)$ is the i'th sub-model weight with batch indexes $\mathcal{B}_t$ and early stopping applied on batch $t-2$. The ensemble model mean is used to represent the "true" model output, as it is expected to be significantly more robust against initialization effects.

Forecasting Evaluation Criteria

To evaluate the ensemble models' forecasting ability throughout trial t, the forecasting evaluation and prediction root mean square error (RMSE) is calculated for trial t and output l by:

$$\mathcal{J}_l(t) = \sqrt{\sum_{p=1}^{N_{s,t-1}-1}\sum_{k=p+1}^{N_{s,t-1}} \frac{\left\|y_l[k \mid t] - \hat{\bar{y}}_l[k \mid t,p]\right\|_2^2}{\frac{1}{2}N_{s,t-1}\left(N_{s,t-1}-1\right)}} \tag{3.27a}$$

$$\mathcal{J}^*_l(t) = \sqrt{\sum_{p=1}^{N_{s,t-1}-1}\sum_{k=p+1}^{N_{s,t-1}} \frac{\left\|y_l[k \mid t] - \hat{\bar{y}}^*_l[k \mid t,p]\right\|_2^2}{\frac{1}{2}N_{s,t-1}\left(N_{s,t-1}-1\right)}} \tag{3.27b}$$

Where $\mathcal{J}_l(t) \in \mathbb{R}_+$ is the evaluation forecasting performance of trial t, $\mathcal{J}^*_l(t) \in \mathbb{R}_+$ is the prediction forecasting performance of trial t, $y_l \in \mathbb{R}$ is the l'th index of $y \in \mathbb{R}^{N_y}$, and the factor $\frac{1}{2}N_{s,t-1}\left(N_{s,t-1}-1\right)$ is a normalization constant. Furthermore, the impact of using substitute future inputs can be quantified by the difference between $\hat{\bar{y}}_l[k \mid t,p]$ and $\hat{\bar{y}}^*_l[k \mid t,p]$, according to

$$\mathcal{JJ}^*_l(t) = \sqrt{\sum_{p=1}^{N_{s,t-1}-1}\sum_{k=p+1}^{N_{s,t-1}} \frac{\left\|\hat{\bar{y}}_l[k \mid t,p] - \hat{\bar{y}}^*_l[k \mid t,p]\right\|_2^2}{\frac{1}{2}N_{s,t-1}\left(N_{s,t-1}-1\right)}}. \tag{3.28}$$

The mean RMSE performance of all trials $t \in \{1,\ldots,N_t\}$, as depicted on Figure 3.7, is then calculated according to

$$\mathcal{M}(\mathcal{J}_l) = \frac{1}{N_t}\sum_{t=1}^{N_t}\mathcal{J}_l(t), \tag{3.29a}$$

$$\mathcal{M}(\mathcal{J}^*_l) = \frac{1}{N_t}\sum_{t=1}^{N_t}\mathcal{J}^*_l(t) \text{ and} \tag{3.29b}$$

$$\mathcal{M}(\mathcal{JJ}^*_l) = \frac{1}{N_t}\sum_{t=1}^{N_t}\mathcal{JJ}^*_l(t), \tag{3.29c}$$

where $\mathcal{M}(\mathcal{J}_l)$, $\mathcal{M}(\mathcal{J}_l^*) \in \mathbb{R}_+$ are the mean evaluation and prediction RMSE performance, and $\mathcal{M}(\mathcal{J}\mathcal{J}_l^*) \in \mathbb{R}_+$ is the RMS evaluation and prediction forecasting difference.

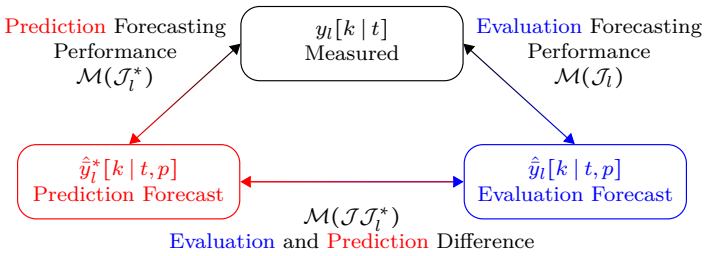


Figure 3.7: Overview of how (3.29) is related to (3.27) and (3.28).

Forecasting Procedure Summary

Forecasting trial t is carried out as follows:

1) Preselect up to N_{PB} batches according to (3.25) from the prior batches $\mathcal{B}_t \subseteq \{1\text{–}N_{PB}, \ldots, t\text{–}3\}$.

2) Formulate $\tilde{u}$ from $\mathcal{B}_t$ and X from $\tilde{u}$ and $\tilde{y}$. From each element in $\tilde{y}$ carry out an IVS as described in Section 3.1 to calculate $\mathcal{S}$ using (3.18).

3) From $\mathcal{S}$ determine the MLP model $\mathcal{N}$ structure governed by $\{\bar{n}, W_{u,i}^h, \bar{m}, W_{y,j}^h\}$ as described in Section 3.2.2, and the inputs u and outputs y using (3.19).

4) Train N_m sub-model weights $\{\mathcal{W}_1, \ldots, \mathcal{W}_{N_m}\}$ with different initial conditions by minimizing (3.22) as described in Section 3.2.3.

5) Calculate the ensemble prediction forecast $\hat{\hat{y}}_l^*[k \mid t, p]$ and evaluation forecast $\hat{\hat{y}}_l[k \mid t, p]$ for all initial samples $p \in [1; N_{s,t-1} - 1]$ and $k \in [p+1; N_{s,t-1}]$ using (3.26).

6) Calculate the mean prediction and evaluation forecasting RMSE $\mathcal{M}(\mathcal{J}_l)$ and $\mathcal{M}(\mathcal{J}_l^*)$ given by (3.27), and the RMS evaluation and prediction forecasting difference $\mathcal{J}\mathcal{J}_l^*(t)$ using (3.28).

A visual overview of prediction forecasting is depicted on Figure 3.8.

3.3.2 Experimental Forecasting

This work is based on data gathered from a ≈20 year old state-of-the-art broiler house located in the northern Denmark. A total of 29 batches are used, which

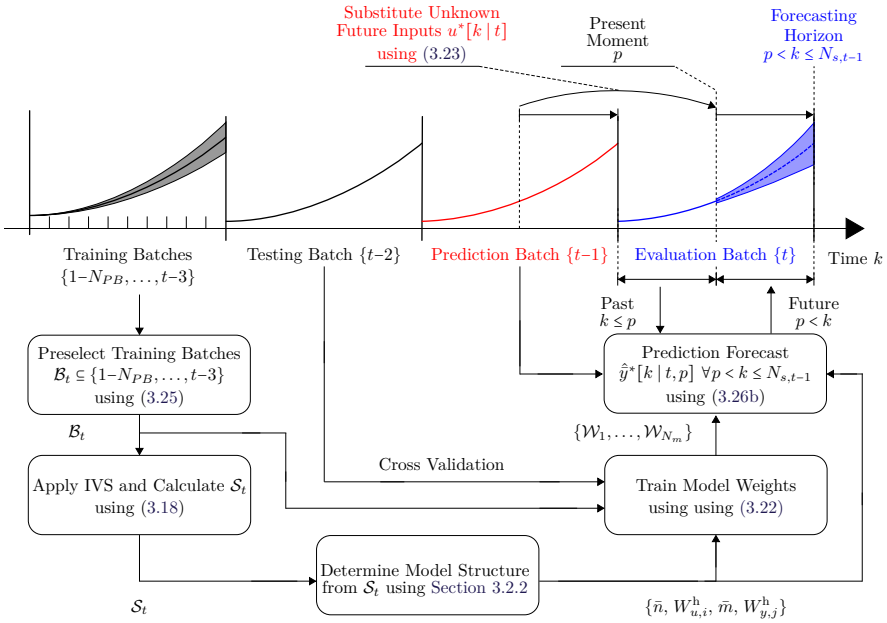


Figure 3.8: Visual overview of prediction forecasting. For evaluation forecasting replace $\hat{y}^*[k \mid t, p]$ from (3.26b) with $\hat{y}[k \mid t, p]$ from (3.26a).

are collected over a period of 3 years and 4 months. Each batch contains 38155 broilers on average with standard deviation 2329.

The potential inputs to the model consist of the available environmental variables for this broiler house: The measured inside temperature, outside temperature and humidity, light intensity reference, ventilation demand, and heating demand. Note that measured CO_2 was not included, as it was only present in 8 of 29 batches. The model outputs consist of the available broiler behavior indicators: the measured weight along with feed and water consumption per bird.

A distinction is made between reference, demand and measured variables. Reference variables are independent, demand variables are determined by a deterministic entity like a controller outside the model, and measured variables are dependent on the model process.

Method Configuration

A sample interval of 3 hours is used with the first sample at midnight (00:00) on the first day regardless of the actual start time of the batch due to technical limitations, which permits the time of day to be partially captured. The mean value of the data within each sample period is used.

Satisfactory results are obtained with $N_m = 64$ sub-models and $N_h = 7$ hidden neurons in the hidden layer of the model. On average this results in 230 free model parameters across all trials. Similarly, good results are obtained with $N_{\mathcal{P}} = 5$ sub-batch training start times at day $\mathcal{P} = \{2, 7, 14, 21, 28\}$ in (3.22).

As the data-quality cannot be guaranteed in broiler production, the number of selected variables and number of lags are deliberately limited – see Section 3.2.2 for more details. A maximum of $N_B = 10$ training and $N_{PB} = 11$ preliminary batches are used – resulting in a total of $N_t = 18$ trials. As the IVS is independently carried out for each of the outputs, a maximum of 18 lagged inputs can be selected. Input and output lags up to $N_{L,\tilde{u}} = N_{L,\tilde{y}} = 16$ samples are considered – equivalent of 2 days. Furthermore, the search space of the IVS, S, is restricted to 3 input classes (e.g. measured temperature) per output where each input class $\tilde{u}$ can have up to 2 lags.

3.3.3 Trial Data Analysis

To better understand the trial results the input and output variables are analyzed, which are depicted on Figure 3.9. Emphasis is put on the difference between what the model "knows" through training data, and the "unknown" validation and prediction data. The IVS results for this trial are depicted on Table 3.2 with rounded rectangles, which wont be commented on in the following.

Input variables

The ventilation demand is negatively correlated with the temperature and humidity. The intuition is that the broilers introduce both heat and water into the air in the broiler house, increasing the temperature and humidity, which are removed through ventilation.

A rule of thumb by the hatchery supplying the day old broilers is to keep the temperature at ≈30 °C in the beginning of the batch and decrement it by 1 °C every 3 days until 20 °C. The beginning temperature appears to be ≈4 °C offset in this case. A low degree of variation in temperature across all batches is noted, compared to e.g. humidity. The overall measured temperature of the prediction- and evaluation batch appear to have similar behavior as the training and testing data. However, the prediction batch is on average ≈1 °C higher onwards of day ≈11.

The heating demand for the prediction and evaluation batch have significantly higher fluctuations onwards of day 15, which are caused by a below average outside temperature. Lastly, note that the light intensity reference data are quite spiky, which is caused by a light program emulating the day and night cycle.

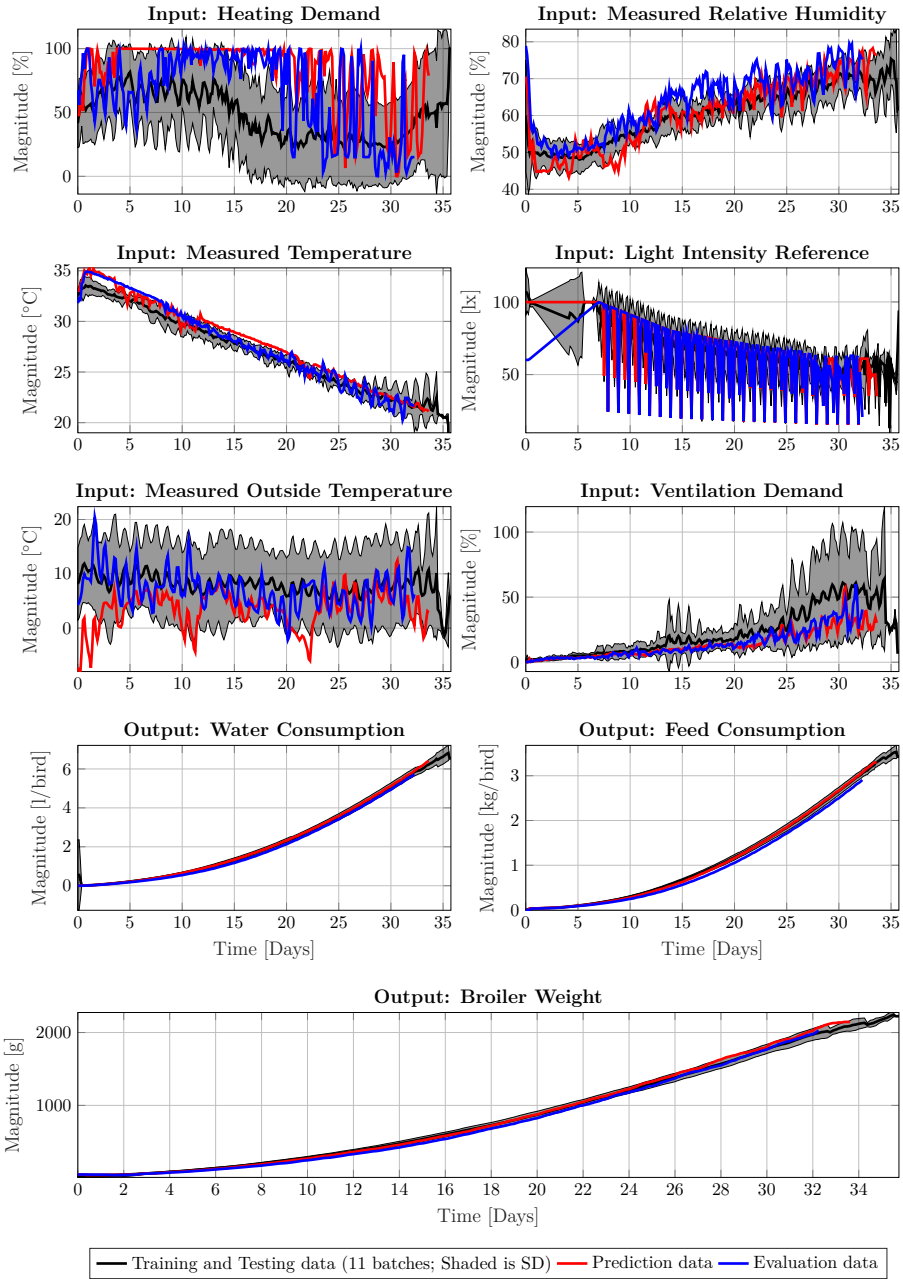


Figure 3.9: The inputs and output data from the batches used in the most recent trial $t = N_t$. The black shaded area denotes one standard deviation from the mean value. Each batch contains 38155 broilers on average with standard deviation 2329.

Output variables

Broiler weight is highly positively correlated with cumulative feed and water consumption, as the two are necessary for both survival and growth of the broilers. Bird weight has been reported by customers to be highly predictable from feed and water consumption – supported by the fact that the three have similar variations and behavior. Note that it is unknown if and when the farmer has changed feed type for any of the batches.

The feed consumption is distinctly lower for the evaluation batch compared to the training, testing and prediction data, but the measured broiler weight and water consumption are within one standard deviation of the training and testing data. For this reason, the prediction forecast is expected to be higher than the evaluation forecast.

3.3.4 Input Variable Selection

In Table 3.2 a summary of all selected inputs, input lags and output lags is depicted. In the following, the selection trends for each output is discussed.

Y		Relevant Potential Input Variables X	Selected Lags S (% selected of all trials)															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Broiler Weight	Input	Measured Relative Humidity	28	22	11	0	17	0	6	22	28	11	17	17	6	6	0	0
		Measured Temperature	0	0	6	6	11	6	6	6	0	0	0	0	0	6	0	0
		Light Intensity Reference	22	0	0	0	0	6	28	0	0	0	0	0	0	0	0	0
		Measured Outside Temperature	6	11	0	11	11	11	17	6	6	11	6	6	0	11	11	11
		Ventilation Demand	17	6	11	22	11	0	11	22	17	0	6	17	6	0	22	11
	Output	Broiler Weight	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Feed Consumption	39	11	6	0	0	6	0	11	11	0	22	11	39	11	17	17
		Water Consumption	44	6	6	0	0	0	11	28	6	28	17	0	22	11	6	17
Feed Consumption	Input	Heating Demand	0	0	22	6	0	0	6	6	6	0	0	17	0	0	0	28
		Measured Relative Humidity	11	11	22	11	6	33	11	17	11	0	11	11	0	0	0	0
		Measured Temperature	11	6	22	6	0	0	0	0	0	0	0	0	0	0	0	0
		Light Intensity Reference	11	0	0	0	0	0	0	11	6	0	0	6	11	11	6	17
		Measured Outside Temperature	0	11	22	11	0	6	11	6	0	6	22	6	11	0	11	22
		Ventilation Demand	17	0	22	0	6	6	11	0	0	0	6	6	11	0	6	0
	Output	Broiler Weight	39	28	17	0	22	0	33	0	6	0	0	0	11	11	17	17
		Feed Consumption	100	61	0	39	0	0	0	0	0	0	0	0	0	0	0	0
		Water Consumption	100	50	0	39	0	0	6	0	6	0	0	0	0	0	0	0
Water Consumption	Input	Heating Demand	0	0	6	0	0	0	6	0	0	0	0	0	0	0	0	0
		Measured Relative Humidity	6	17	6	6	0	11	11	6	0	0	6	6	6	11	0	6
		Measured Temperature	11	0	6	6	0	0	0	0	0	0	0	0	0	0	6	0
		Light Intensity Reference	44	0	6	0	0	0	0	44	11	0	0	0	0	0	0	6
		Measured Outside Temperature	0	6	6	0	0	6	6	0	0	0	6	17	6	17	17	0
		Ventilation Demand	39	0	17	0	0	11	11	11	0	6	28	6	0	0	0	6
	Output	Broiler Weight	0	6	11	0	0	28	17	17	6	22	6	6	0	22	0	39
		Feed Consumption	61	0	22	0	0	0	17	11	0	6	22	0	6	0	0	0
		Water Consumption	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.2: Overview of the selected inputs, input lags and output lags for each of the N_t =18 trials. A separate IVS of the selected lags X resulting in the selected lags S is presented for each output Y. The lags used in the most recent trial is marked with a solid black bars, while trends are marked with shaded gray bars. Note that Light Intensity Reference was only present in 12/18 tests.

Broiler weight

The primarily selected inputs for the broiler weight are the measured relative humidity, measured outside temperature and ventilation demand. The measured relative humidity appears to have a short and long horizon with lags at 0-2 and 7-11 – equivalent of "now" and a day ago (8 samples per 24 hour). The measured outside temperature appears to have no clear trends in the selected lags, which suggest a seasonal dependence rather than a dependence on outside temperature. The ventilation demand also appears to have no clear trends, but an argument for small clusters at around lag 0, 3, 7, 11 and 14 can be made with a periodicity of ≈4 lags (12 hours).

The selected outputs for the broiler weight appear to be strongly correlated with lag 0 and 1 of itself. It also appears to be dependent on the two other outputs, either very recently at lag 0 and 1 or after lag 7.

Feed Consumption

The primary selected inputs for the feed consumption are the measured relative humidity and measured outside temperature. The measured relative humidity appears to be selected within the first 8 lags. Like the broiler weight input selection, the measured outside temperature appears to have no clear trends.

The selected outputs for the feed consumption are primarily between lag 0 and 3 for both the feed consumption and water consumption. Broiler weight appears to have a short term component between lag 0 and 6 and a long term component between lag 12-15.

Water Consumption

The primary selected inputs for the water consumption are the ventilation demand, light intensity reference and measured relative humidity. The ventilation demand appears to have a weak trend between lag 0 and 11 – an argument can be made for small clusters at lag 0, 2, 6 and 10. The light intensity reference has a strong trend at lag 0 and 7, indicating a day rhythm as they are spaced ≈24 hours apart. The measured relative humidity appears to have no strong trends, but it is likely that there are small clusters at lag 1, 5, 12 and 15.

The selected outputs for the water consumption appear to be strongly correlated with lag 0 and 1 of itself, like the broiler weight output. Broiler weight appears to have a long term component between lag 5 and 15, possibly with a small cluster at lag 7, 13 and 15. Feed consumption appears to have two clusters, one between lag 0 and 2, and one between lag 6 to 10.

Other Trends

The measured temperature was not primarily selected for any of the outputs. This might suggest that the temperature is in an optimum range for the ma-

jority of the batches, as temperature has a strong influence on broiler growth. Alternatively, this might be due to it being tightly controlled.

Heating demand was not primarily used for any of the outputs, but comes quite close in feed consumption. Hence, heating demand might contribute little to broiler forecasting.

3.3.5 Forecasting Case Study

A detailed forecasting case study of the most recent trial, $t = N_t$, is presented in this section.

Weight Forecasting

Both the prediction and evaluation forecasts show good overall forecasting capability, in the sense that it tracks the measured output shape throughout the batch – however, the model has a tendency to underestimate for all outputs. Comparing the prediction and evaluation forecasts, $\hat{\hat{y}}_l[k \mid t, p] - \hat{\hat{y}}_l^*[k \mid t, p]$, a difference of −8g broiler weight, 87g feed and 62ml water are observed – essentially predicting a similar broiler weight for less feed and water. Comparing this with the measured data of the prediction- and evaluation batch on Figure 3.9 at the end of the evaluation batch, the evaluation batch produces the same amount of meat with 3.8% less feed and 1.1% less water, which supports the previous observation. This indicates that the model captures at least some of the dynamic interconnection between the two forecasts, as the climate conditions are the only difference between the prediction and evaluation forecasts.

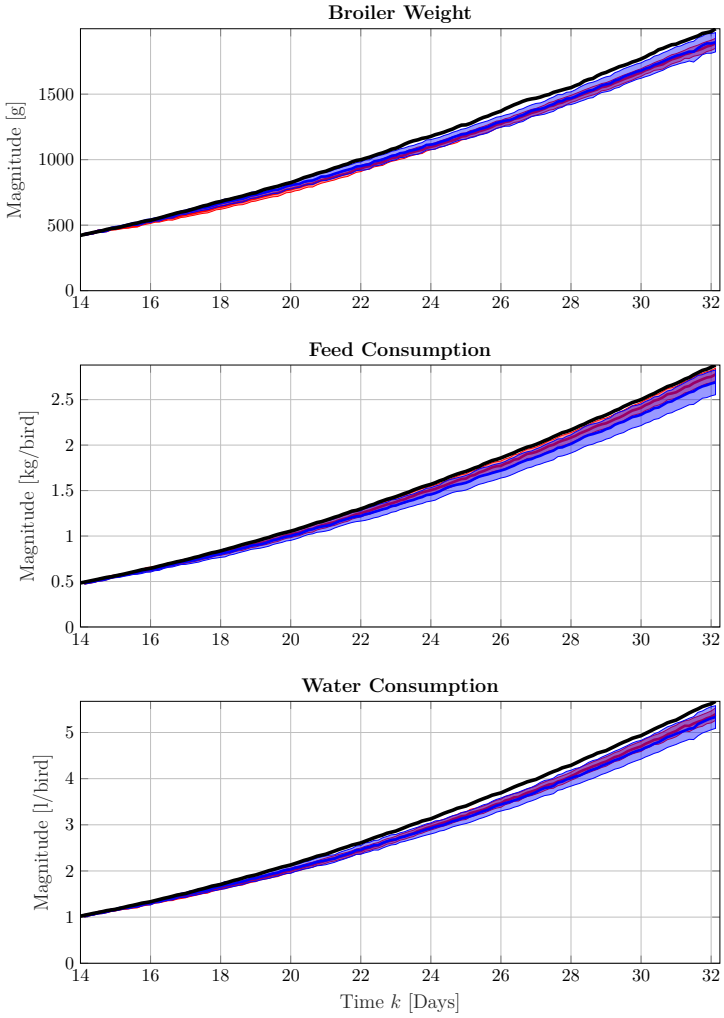
Weight on Day Forecasting

Currently, the broiler industry is particularly attuned to the broiler weight on particular samples, e.g. samples equivalent of day 7, 14, 21 and 28, which among others are used to determine when a batch is expected to be ready for slaughter. Therefore the models' ability to forecast the output at these days from all prior samples is investigated, which is denoted *weight on day forecasting*. The model output sample k is fixed, e.g. k = day 28, and the forecasting horizon is determined through the variable initialization sample p, i.e. p = day 7, for $p < k$. Note that this differs from traditional forecasting, where p is fixed and k is variable.

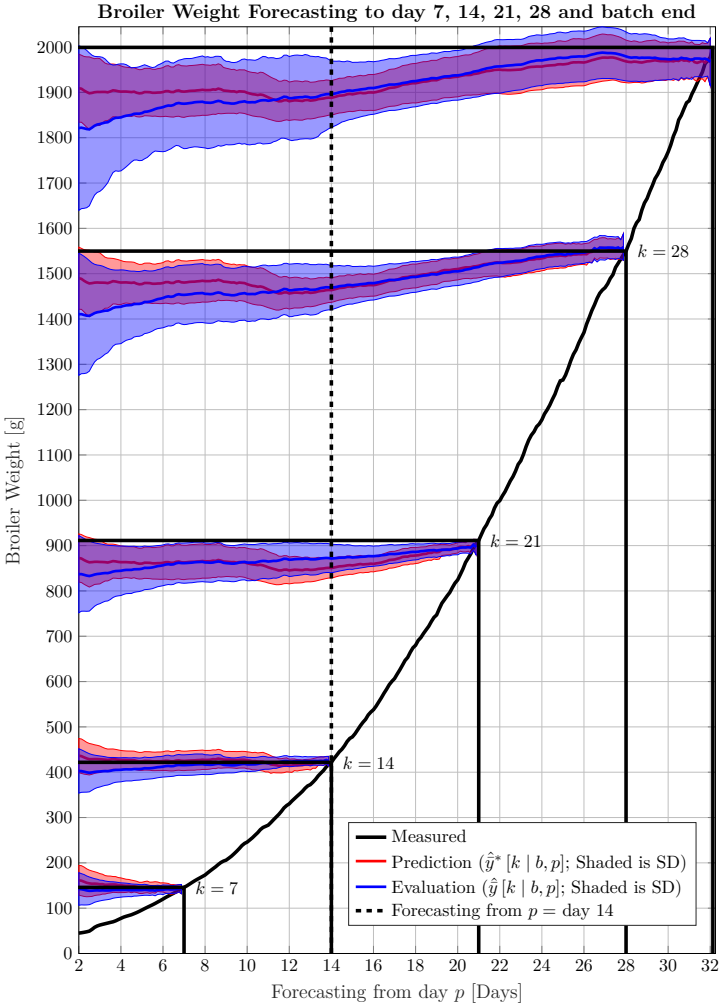
Weight on day forecasting for trial $t = N_t$ is depicted on Figure 3.10b. The under-estimation trend is present when forecasting weight after day 14, as the weight on day 7 and 14 forecasts show good agreement with the measured weight. The weight on day 21 and 28 forecasts are underestimated when forecasting from before day ≈22, which is caused by a higher than average growth in this period. To elaborate, between day ≈7 to ≈24 the broiler weight is more than one standard deviation lower than the training and testing data, but it is close to the mean broiler weight after day ≈24 – the ensemble model cannot

explain this from the input data. This could be caused by biased broiler weight measurements in this interval.

The evaluation and prediction forecasts resemble each other well in terms of ensemble model standard deviation as the shaded area is similar for the two forecasts, indicating that substitute input data from the previous batch can be used to represent the ensemble standard deviation of the current batch despite different ensemble model means.



(a) Forecasting from p = day 14 to batch end. See Section 3.3.1 for details.



(b) Broiler Weight on day forecasting to k = day 7, 14, 21, 28 and batch end. See Section 3.3.1 for details.

Figure 3.10: The solid blue evaluation forecasts are generated with models initialized with evaluation batch data and future inputs from the evaluation batch. The solid red prediction forecasts are initialized with evaluation batch data, but with substitute future inputs from the prediction batch. The blue traces use future input data from the current batch, and the red traces use substitute future input data from the previous batch. The blue and red bold lines are the ensemble model means, while the shaded areas denote the ensemble standard deviation. Figure 3.10a depicts a single forecast from day p = 14 to the batch end. Figure 3.10b aims at showing the weight forecast of Figure 3.10a for all start samples, p, on specific days, k, marked with vertical solid black lines. The vertical dashed line on Figure 3.10b marks the forecast on Figure 3.10a. The horizontal solid black lines show the measured weight for that day.

3.3.6 Trial Performance Evaluations

In this section the RMS forecasting error depicted on Figure 3.11 and Table 3.3 is analyzed. From Figure 3.11 no pattern for broiler weight and feed consumption is present, but an argument could be made for a seasonal influence for the water consumption with a period of ≈6 trials – equivalent to ≈1 year. This is not a source of concern, as the main objective is broiler weight forecasting. Most importantly, $\mathcal{J}^*$ appears to consistently track the trend of $\mathcal{J}$, suggesting that substituting unknown future inputs with inputs from a previous batch does not degrade performance significantly. From the mean RMS forecasting errors from Table 3.3, it can be concluded that $\mathcal{M}(\mathcal{J}\mathcal{J}^*) \neq 0$ and $\mathcal{M}(\mathcal{J}^*) \approx \mathcal{M}(\mathcal{J})$ for all outputs – see Figure 3.7 for clarification. *This strongly suggests that using substitute future inputs in forecasting does not deteriorate forecasting performance significantly, while simultaneously suggesting that the broiler house conditions have a significant impact on the outputs*, e.g., a RMSE difference of 2.7g in the case of broiler weight forecasting. In conclusion, the method is shown to be practically applicable to broiler weight forecasting and the achieved mean forecasting RMSE of 66.8g is acceptable in state-of-the-art broiler industry.

Output	Mean Forecasting RMSE		
	$\mathcal{M}(\mathcal{J}^*)$	$\mathcal{M}(\mathcal{J})$	$\mathcal{M}(\mathcal{J}\mathcal{J}^*)$
Weight [g]	66.8	64.1	35.5
Feed [g/bird]	90.2	87.8	63.6
Water [ml/bird]	185.8	169.2	133.2

Table 3.3: The mean prediction and evaluation forecasting RMSE along with their difference are calculated using (3.29).

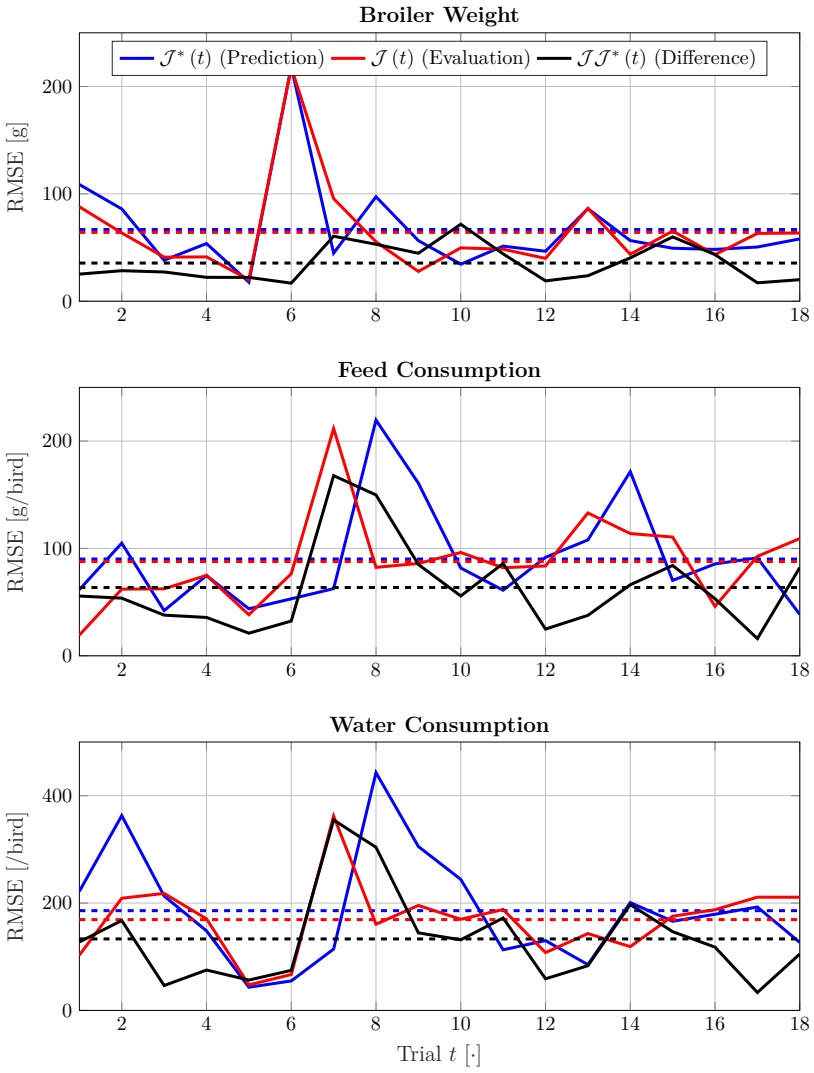


Figure 3.11: Prediction and evaluation RMS forecasting error calculated using (3.27) and (3.28) of all trials, where a dashed line is the mean from Table 3.3.

3.4 Improved Slaughter Weight Forecasting

Automatic weighing pads are commonly used for weighing broilers, which require one or more broilers to physically stand on the pad to be measured. The measured broiler weight is known to be negatively biased onwards of day 15-21, although the exact mechanism behind this bias is not fully understood, which has been partially corrected by introducing a heuristic time-dependent correction factor – as discussed in Section 2.1.2. Note that a standard deviation between the measured local weight and the slaughter weight of more than 100g, equivalent of 5% of the total slaughter weight, is not uncommon.

The slaughter weight is considered very accurate and is included by overriding the last measured local weight at sample $k = N_{s,b}$ of each batch. Extra emphasis is then placed on the slaughter weight at sample $k = N_{s,b}$ in the cost function, while samples onwards of $N_\phi \in \mathbb{Z}_+$ are weighted less. A function achieving the desired behavior is given by

$$\phi(k) = \begin{cases} 1, & k < N_\phi \\ 1 + (N_{s,b} - N_\phi)(\gamma - 1), & k = N_{s,b} \\ \gamma, & \text{otherwise} \end{cases} \tag{3.30}$$

where $\phi : \mathbb{Z}_+ \to \mathbb{R}$ is the cost shaping function, $N_\phi \in \mathbb{Z}$ is the start cost shaping sample number, and $\gamma \in]0; 1[$ is the cost shaping weight. On Figure 3.12 an example of (3.30) is depicted. Note that $\mathcal{P}_p \leq N_\phi$ for all p is required to avoid changing the relative weighting of other variables, such as feed consumption, as

$$\frac{1}{N_{s,b} - \mathcal{P}_p + 1} \sum_{k=\mathcal{P}_p}^{N_{s,b}} \phi(k) = \frac{1}{N_{s,b} - \mathcal{P}_p + 1} \sum_{k=\mathcal{P}_p}^{N_{s,b}} 1 = 1.$$

Rewriting (3.22) to include the slaughter weight and the weight shaping function $\phi(k)$ given by (3.30), results in

$$\mathcal{W}^\star(\mathcal{B}, c, N_\phi, \gamma) = \arg\min_{\tilde{\mathcal{W}}} \sum_{b \in \mathcal{B} \setminus c} \frac{E_b^*(\tilde{\mathcal{W}}, N_\phi, \gamma)}{\#\mathcal{B} - 1} \tag{3.31}$$

with

$$E_b^*(\tilde{\mathcal{W}}, N_\phi, \gamma) = \sum_{p=1}^{N_\mathcal{P}} \sum_{k=\mathcal{P}_p}^{N_{s,b}} \frac{\mathcal{E}_b}{N_{r,b}} + \bar{\alpha} \|\tilde{\mathcal{W}}\|^2$$

$$\mathcal{E}_b = \sum_{i=1}^{N_y} \begin{cases} \|\Gamma_b - \hat{y}_i\|_2^2 \, \phi(k), & k = N_{s,b} \wedge i = i_w \\ \|y_i - \hat{y}_i\|_2^2 \, \phi(k), & k \neq N_{s,b} \wedge i = i_w \\ \|y_i - \hat{y}_i\|_2^2, & \text{otherwise} \end{cases}$$

where $y_i = y_i[k \mid b]$ is the target output, $\hat{y}_i = \hat{y}_i[k \mid \mathcal{P}_p, b, \tilde{\mathcal{W}}]$ is the model output, Γ_b is the slaughter weight of batch b, and i_w is the index of the measured weight.

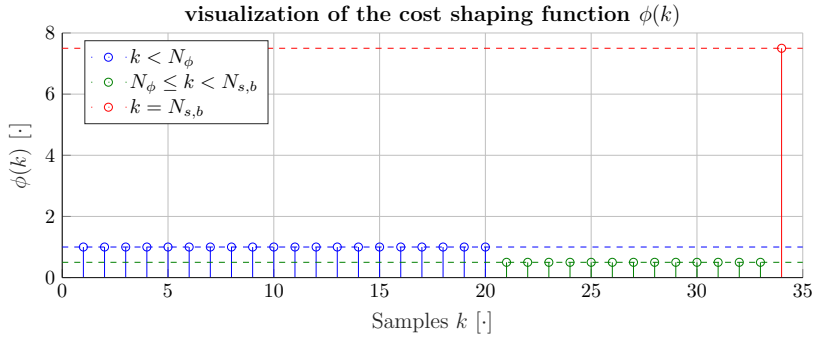


Figure 3.12: Visualization of the cost shaping function $\phi(k)$ with $N_\phi = 20$, $N_{s,b} = 34$ and $\gamma = 0.5$. The blue, green and red values correspond to a separate case of (3.30).

The ensemble forecasting model mean of trial $t \in \mathbb{Z}_+$ and output $l \in \mathbb{Z}_+$ for the two algorithms are given by

$$\hat{\bar{y}}_l[k \mid t, p] = \sum_{i=1}^{N_m} \frac{\hat{y}_l[k \mid t, p, \mathcal{W}_i(\mathcal{B}_t, \min\{\mathcal{B}_t\})]}{N_m} \text{ and} \tag{3.32a}$$

$$\hat{\bar{y}}^*_l[k \mid t, p, N_\phi, \gamma] = \sum_{i=1}^{N_m} \frac{\hat{y}_l[k \mid t, p, \mathcal{W}^*_i(\mathcal{B}_t, \min\{\mathcal{B}_t\}, N_\phi, \gamma)]}{N_m}, \tag{3.32b}$$

where $\hat{\bar{y}}_l[k \mid t, p]$ and $\hat{\bar{y}}^*_l[k \mid t, p, N_\phi, \gamma]$ are the ensemble forecasting mean of the nominal and improved algorithm, $\mathcal{W}_i(\mathcal{B}_t, \min\{\mathcal{B}_t\})$ is the i'th trained sub-model weight using the nominal algorithm (3.22), and $\mathcal{W}^*_i(\mathcal{B}_t, \min\{\mathcal{B}_t\}, N_\phi, \gamma)$ is trained using the improved algorithm (3.31). The cross validation batch or testing batch, c, is determined as $\min\{\mathcal{B}_t\}$ instead of $t - 2$.

3.4.1 Experimental Results

The data is from the same location as Section 3.3, but it includes the measured slaughter weight in addition to 6 more batches – corresponding to a total of more than 4.5 years of production data.

On Figure 3.13, the locally measured weight is compared with the measured slaughter weight. In this case, the local measured weight has been quite nicely corrected with the previously mentioned correction factor, as the mean error is quite low. However, a standard error of 115.9g is quite significant, which suggests that there is still room for improvements as the "behavior" constant has not corrected the biased weight problem completely.

Note that compared to Section 3.3, both the heating rate and ventilation rate are not included in this work, as excluding them results in more accurate forecasts. This is caused by a mismatch between the future substituted inputs from the prediction batch and the actual future values, sometimes resulting in highly unrealistic scenarios.

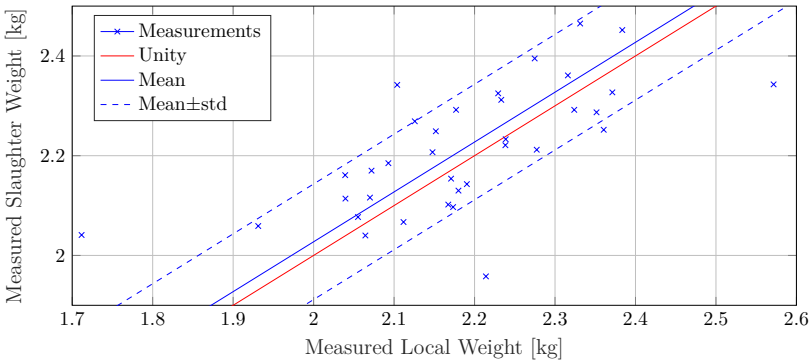


Figure 3.13: Comparison between measured local weight and slaughter weight, with mean difference of 27.4g with a standard deviation of 115.9g, and median error of 45g.

Default Algorithm Configuration

The default configuration from Section 3.3 is unaltered unless specified, with the aim of making the results somewhat comparable. It uses $N_B = 10$ training batches, $N_m = 64$ sub-models, $N_h = 7$ neurons in the hidden layer, and $N_{\mathcal{P}} = 5$ sub-batch training start times at day $\mathcal{P} = \{2, 7, 14, 21, 28\}$. The sample interval is changed from $T_s = 3$ hours in Section 3.3 to $T_s = 12$ hours to reduce training time. In Section 3.3, 3 input variables were selected, which is reduced to 2 as fewer input candidates are used.

Test Evaluation Criteria

The objective of this study is to investigate the proposed algorithm's slaughter weight forecasting ability. The forecasting RMSE of all N_t tests between the slaughter weight Γ_t at sample $N_{s,t}$ and the weight forecast at sample $N_{s,t}$ from the prior $\Delta \in \mathbb{Z}_+$ samples, i.e. $p \in \{N_{s,t} - \Delta, \ldots, N_{s,t} - 1\}$, is defined by

$$\hat{\bar{\mathcal{J}}}_\Delta = \sqrt{\sum_{t=1}^{N_t} \sum_{p=N_{s,t}-\Delta}^{N_{s,t}-1} \frac{\left(\hat{\bar{y}}_{i_w}[N_{s,t} \mid t, p] - \Gamma_t\right)^2}{N_t \, \Delta}} \tag{3.33a}$$

$$\hat{\bar{\mathcal{J}}}^*_\Delta = \sqrt{\sum_{t=1}^{N_t} \sum_{p=N_{s,t}-\Delta}^{N_{s,t}-1} \frac{\left(\hat{\bar{y}}^*_{i_w}[N_{s,t} \mid t, p, N_\phi, \gamma] - \Gamma_t\right)^2}{N_t \, \Delta}} \tag{3.33b}$$

where $\hat{\bar{\mathcal{J}}}_\Delta$ is the forecasting RMSE of the nominal algorithm and $\hat{\bar{\mathcal{J}}}^*_\Delta$ is the RMSE of the improved algorithm. Hence, it is possible to investigate forecasting performance at different forecasting horizons, e.g. a 7 day horizon with a sample interval T_s of 12 hours is obtained with $\Delta = 14$. Furthermore, the RMSE

between the measured local weight and the slaughter weight is calculated by

$$\mathcal{J}_0 = \sqrt{\sum_{t=1}^{N_t} \frac{\left(y_{i_w}[N_{s,t} \mid t] - \Gamma_t\right)^2}{N_t}}, \tag{3.34}$$

which allows for comparison of the algorithm with the locally measured weight used for training prior to slaughter.

Test Description

In the following, several tests are specified with the intention of providing insights into effective configuration of the improved slaughter weight forecasting algorithm. Each test explores the effect of one or two parameters by changing it from the default configuration described in Section 3.4.1.

Test #1 investigates the effect of including the slaughter weight through the weight shaping function as detailed in Section 3.4. The measured local weight is used as a baseline through $\mathcal{J}_1$ with $\Delta = 1$. The statistical significance of using the improved algorithm is determined using an ANOVA-factor analysis with 5% significance level of the residuals, i.e., $\hat{\bar{y}}^{\star}_w - \Gamma$. This test assumes normally distributed variables, which are checked using the Anderson-Darling normal distribution test with a 5% significance level.

Test #2 investigates different settings of the weight shaping parameters N_ϕ and γ in (3.30). Setting $N_\phi = \text{day } \{12, 15, 18, 21\}$, the aim is to determine the bias onset – currently, it is believed to be around day 15-21. Setting $\gamma = \{0, 0.1, 0.2\}$, the aim is to investigate the influence of γ.

Test #3 investigates the effect of different number of training batches N_B, and subsequently the parameter-drift rate. $N_B = 10$ training batches has been heuristically chosen – corresponding to ≈1.5 years of production data.

Test #4 investigates the number of hidden neurons N_h, which has been heuristically chosen to be $N_h = 7$.

Test #5 investigates sample interval T_s, which has been heuristically chosen to be 3 hours.

Note that multiple test-iterations can be combined to form a "steepest descent"-like parameter optimization algorithm. Such an approach has only been carried out for the two new parameters N_ϕ and γ in the following. An exhaustive search is not practically feasible, as the test suite takes around 33 hours to complete.

Test Results

The test results of the five test cases described in Section 3.4.1 are depicted on Table 3.4.

Test #1 On Figure 3.14 a box plot of the slaughter weight prediction errors for Test #1 with $\Delta = 1$ is depicted. It shows that the median measured weight error is 50g with an interquartile range of 145g – suggesting that the measured weight has a large variation. The nominal algorithm has a median weight error of -154g and interquartile range of 100g – suggesting that the nominal algorithm is highly negatively biased for slaughter weight forecasting. The modified algorithm has a median weight error of -8g and interquartile range of 96g – suggesting that using the slaughter weight greatly increases the accuracy of the algorithm but not necessarily the precision. Comparing the RMSE for the most recent sample ($\Delta = 1$) and last 7 days ($\Delta = 14$) listed in Table 3.4 shows that the improved algorithm is superior to both the measured local weight and the nominal algorithm.

The Anderson-Darling test cannot reject that any of the residuals with $\Delta = 1$ of the three groups for Test #1 are normally distributed at a 5% significance level. According to the ANOVA factor analysis, the means of all groups are significantly different at a 5% significance level. The means of the nominal and improved method is significantly different with a p-value of 0.044, which might be susceptible to change with growing sample sizes.

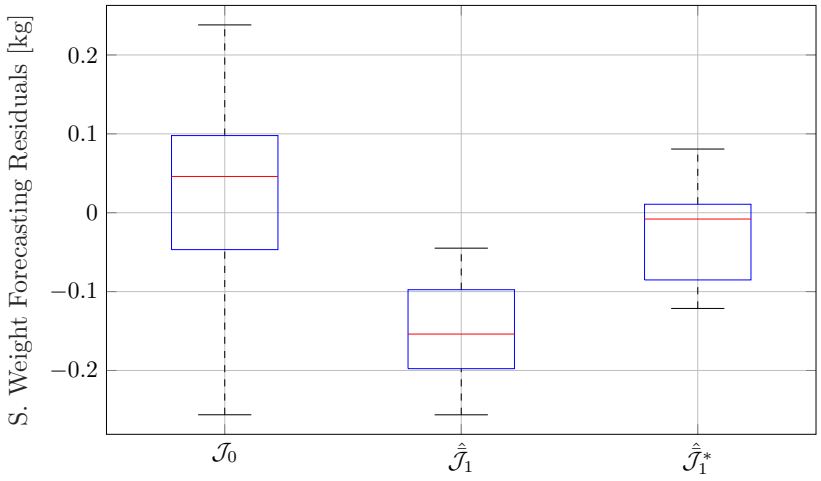


Figure 3.14: Box plot of Test #1 residuals of the measured weight $\mathcal{J}_0$, and both the nominal algorithm $\hat{\mathcal{J}}_1$ and improved algorithm $\hat{\mathcal{J}}_1^*$ with $\Delta = 1$. See Section 3.4.1 for details.

	Type	**Last Sample ($\Delta = 1$)**	**Last 7 Days ($\Delta = 14$)**
Test #1	$\mathcal{J}_0$	🟡 106.5	🟡 106.5
	$\hat{\hat{\mathcal{J}}}_\Delta$	🔴 162.4	🔴 167.1
	$\hat{\hat{\mathcal{J}}}_\Delta^{*\dagger}$	🟢 65.4	🟢 73.6

	N_ϕ [Day]	γ [·]	$\hat{\hat{\mathcal{J}}}_1^*$ [g]	$\hat{\hat{\mathcal{J}}}_{14}^*$ [g]
Test #2	12	0	🟢 75.0	🟢 76.7
	12	0.1	🟠 90.0	🟡 89.6
	12	0.2	🟠 90.8	🟡 88.5
	15†	0†	🟢 65.4	🟢 73.6
	15	0.1	🟡 83.2	🟠 91.1
	15	0.2	🔴 93.4	🟠 90.8
	18	0	🟢 71.1	🟢 76.2
	18	0.1	🟡 85.4	🟡 87.3
	18	0.2	🟠 89.2	🟠 91.8
	21	0	🟢 77.6	🟢 83.6
	21	0.1	🟠 88.6	🔴 94.0
	21	0.2	🔴 95.6	🔴 96.6

	N_B [·]	$\hat{\hat{\mathcal{J}}}_1^*$ [g]	$\hat{\hat{\mathcal{J}}}_{14}^*$ [g]
Test #3	14	🟡 76.0	🟡 86.4
	12	🟢 66.2	🟢 69.7
	10†	🟢 65.4	🟢 73.6
	8	🟡 77.7	🟡 84.6
	6	🟢 65.3	🟢 77.0
	4	🔴 116.8	🔴 150.5

	N_h [·]	$\hat{\hat{\mathcal{J}}}_1^*$ [g]	$\hat{\hat{\mathcal{J}}}_{14}^*$ [g]
Test #4	11	🔴 73.0	🔴 79.1
	9	🟠 72.0	🔴 78.6
	7†	🟢 65.4	🟡 73.6
	5	🟢 66.2	🟢 72.2
	3	🟡 71.4	🟢 71.7

	T_s [Hours]	$\hat{\hat{\mathcal{J}}}_1^*$ [g]
Test #5	24	🔴 90.7
	12†	🟢 65.4
	6	🟡 70.8
	3	🟢 65.0

Table 3.4: Experimental test results. The colored circles denote the relative test score, with green 🟢 being the best and red 🔴 the worst performing, while † denotes the default configuration. Note that Δ is the forecasting horizon in (3.33).

Test #2 The best performing weight shaping parameters are $N_\phi = 15$ and $\gamma = 0$. The best performing configurations with fixed N_ϕ uses $\gamma = 0$, which makes sense as it avoids discontinuous jumps between the slaughter weight and the measured weight during training. If the shape of $\phi(k)$ was changed to monotonically decrease from 1 to 0 between N_ϕ and $N_{s,t}-1$, then better performance might be obtained. Similarly, the best performing configurations with fixed γ uses $N_\phi = 15$, where a lower or higher value results in decreased performance. This coincides nicely with the previously stated heuristic claim that the measured broiler weight is biased onwards of day 15.

Test #3 The best choice of number of batches with $\Delta = 1$ and $\Delta = 14$ are $N_\phi = \{6, 10, 12\}$. As $N_B = 4$ produces significantly worse results than $N_B = 6$, it is recommended to use more than 4 batches for training. Similarly, given that $N_B = 14$ and $N_B = 8$ produce worse results, a N_B around $\{10, 12\}$ is recommended because they consistently have good $\hat{\bar{\mathcal{J}}}_1^*$ and $\hat{\bar{\mathcal{J}}}_{14}^*$. However, as $N_B = 6$ has good performance, the valid range is expected to be between 6 and 12. This result agrees well with the heuristic parameter drift claim discussed in Section 3.4.1.

Test #4 The best choice of number of hidden neurons are $N_h = \{5, 7\}$, where $N_h = 5$ is better with a very short forecasting horizon and $N_h = 7$ is better for a longer forecasting horizon. Due to a similar performance it could be advantageous to use $N_h \approx 5$, as it reduce the maximal number of model parameters in $\mathcal{W}$ from 248 to 178, before input selection is applied to reduce the input space.

Test #5 According to the test, the best choice of sampling interval are $T_s = \{3, 12\}$ hours. Given a huge training time difference between the two, a sampling interval of $T_s = 12$ hours is preferred. However, a sampling interval of $T_s = 24$ hours is not recommended.

4 Broiler Feed Conversion Rate Optimization

This chapter provides an overview of the data driven broiler feed conversion rate (FCR) optimization efforts detailed in Paper D, Paper E and Paper F.

4.1 FCR Optimization using Iterative Learning Control

4.1.1 Iterative Learning Control

In super-vector notation, consider the plant with the tracking error given by

$$Y_k(U_k) = P_k U_k + K_k \text{ and} \tag{4.1a}$$
$$E_k(U_k) = R - Y_k(U_k), \tag{4.1b}$$

with

$$Y_k = \begin{bmatrix} y_k[N_s]^T & \cdots & y_k[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_Y}, \tag{4.2a}$$
$$U_k = \begin{bmatrix} u_k[N_s]^T & \cdots & u_k[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_U} \text{ and} \tag{4.2b}$$
$$R = \begin{bmatrix} r[N_s]^T & \cdots & r[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_Y}, \tag{4.2c}$$

where $(\cdot)_k$ denotes the k'th trial, N_s is the start sample, and N_e is the end sample. $Y_k \in \mathbb{R}^{N_Y}$, $U_k \in \mathbb{R}^{N_U}$, $R \in \mathbb{R}^{N_Y}$ and $E_k : \mathbb{R}^{N_U} \to \mathbb{R}^{N_Y}$ are the super-vector output, input, reference and tracking error. $y_k[n] \in \mathbb{R}^{N_y}$, $u_k[n] \in \mathbb{R}^{N_u}$ and $r[n] \in \mathbb{R}^{N_y}$ are the n'th output, input and reference. Note that $N_Y = N_y(N_e - N_s + 1)$ in the case of Y_k. $P_k \in \mathbb{R}^{N_Y \times N_U}$ is the super-vector system matrix, and $K_k \in \mathbb{R}^{N_Y}$ is effects unrelated to the input U_k.

ILC aims at iteratively finding the next input sequence U_{k+1} such that the output sequence Y_{k+1} converges to the desired reference sequence R in k, corresponding to

$$\lim_{k \to \infty} Y_{k+1}(U_{k+1}) = R. \tag{4.3}$$

Input constrained norm optimal ILC is used to calculate the next ILC input U_{k+1} by solving

$$\begin{aligned} U_{k+1} = \arg\min_{U} \; & \|E_k(U)\|^2_{W_E} + \|U - U_k\|^2_{W_{\Delta U}} \\ \text{s.t.} \quad & C_l \leq U - U_k \leq C_u \end{aligned} \tag{4.4}$$

where $\|x\|_Q = \sqrt{x^T Q x}$ is the weighted euclidean norm, $U \in \mathbb{R}^{N_U}$ is the optimization variable, $C_l \in \mathbb{R}^{N_U}$ and $C_u \in \mathbb{R}^{N_U}$ are the lower and upper bounds on

the ILC input change relative to the last input U_k, $W_E \in \mathbb{R}^{N_Y \times N_Y}$ is the output error cost matrix, and $W_{\Delta U} \in \mathbb{R}^{N_U \times N_U}$ is the input change cost matrix.

In the following, (4.4) is formulated as a quadratic minimization problem. Rewriting the tracking error in terms of the input change, conveniently denoted $\Delta U = U - U_k$, results in

$$E_k(U) = R - P_k\left[U_k + \Delta U\right] - K_k = E_k(U_k) - P_k \Delta U.$$

Grouping the cost function in (4.4) by ΔU and $E_k(U_k)$ yields

$$\|\Delta U\|^2_{\left(P_k^T W_E P_k + W_{\Delta U}\right)} + \|E_k(U_k)\|^2_{W_E} - 2\, E_k(U_k)^T W_E P_k \Delta U,$$

which can be reformulated as a quadratic minimization problem with constraints by

$$\begin{aligned} U_{k+1} = U_k + \underset{\Delta U}{\arg\min}\ & \tfrac{1}{2}\|\Delta U\|^2_{Q_1} + Q_2^T \Delta U \\ \text{s.t.}\quad & C_l \le \Delta U \le C_u \end{aligned} \tag{4.5}$$

with

$$\begin{aligned} Q_1 &= 2P_k^T W_E P_k + 2W_{\Delta U} \text{ and} \\ Q_2 &= -2P_k^T W_E E_k(U_k). \end{aligned}$$

This problem can be solved by standard quadratic programming algorithms, such as Matlab's `quadprog` implementation. To get an idea of the solution of (4.5) without constraints, the norm optimal ILC input change ΔU is solved for in

$$0 = \frac{d\left(\frac{1}{2}\|\Delta U\|^2_{Q_1} + Q_2^T \Delta U\right)}{d\Delta U} = Q_1 \Delta U + Q_2,$$

resulting in the optimal solution given by

$$\Delta U = -Q_1^{-1} Q_2 = \overbrace{\left(P_k^T W_e P_k + W_{\Delta U}\right)^{-1} P_k^T W_E}^{L_k} E_k(U_k),$$

where $0 < P_k^T W_E P_k + W_{\Delta U}$ of which $0 < W_{\Delta U}$ and $0 < W_E$ are imposed. Note that apart from the definiteness, no particular structure is imposed on W_E and $W_{\Delta U}$, hence, a different weight can be assigned to each sample.

The unconstrained solution of (4.5) has the shape of

$$U_{k+1} = U_k + L_k E_k(U_k), \tag{4.6}$$

and is known as an Arimoto type ILC feedback law. It is illustrated on Figure 4.1. According to [Xu, 2011], this relatively simple first order control law

is adequate, even for highly nonlinear and uncertain systems. Assuming static P, K and L, then

$$\begin{aligned} E_{k+1}(U_{k+1}) &= R - Y_{k+1}(U_{k+1}) = R - P\,U_{k+1} - K \\ &= R - P\,U_k - K - P\,L\,E_k(U_k) \\ &= (I - PL)\,E_k(U_k). \end{aligned} \tag{4.7}$$

Hence, to design an asymptotically stable ILC controller in the trial domain k, the eigenvalues of the matrix $(I - PL)$ must be inside the unit circle. The fastest converging controller is obtained with $L = P^{-1}$ [sung Ahn et al., 2006a], which in the case of norm optimal ILC is obtained with $W_E = I$ and $W_{\Delta U} = 0$. This makes intuitive sense as $E_{k+1}(U_k) = \left(I - PP^{-1}\right)E_k(U_k) = 0$, i.e., it converges in a single trial. Practical issues often arise when calculating P^{-1}, as P is often ill conditioned, large and non-invertible, which norm optimal ILC does not suffer from to the same extend due to the regularization effect of the weighting matrices W_E and $W_{\Delta U}$.

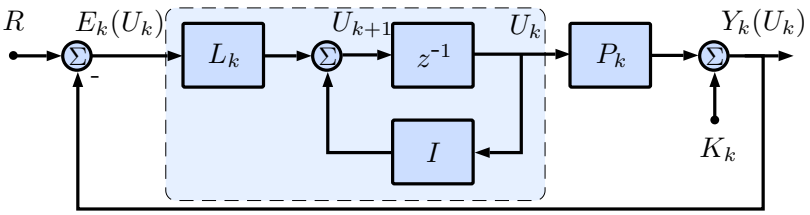


Figure 4.1: A graphical illustration of (4.1) and (4.6), where the ILC governed by (4.6) is highlighted for clarity.

Dynamic Neural Network Super-Vector Model

The system matrix P_k required for calculation of (4.5) is calculated using the data driven DNN model in Section 3.2 (Dynamic Neural Network). The training trials are selected according to Figure 4.2 by

$$\begin{aligned} \mathcal{B}_k = \arg\max_{\tilde{\mathcal{B}}} \; & N_{\tilde{u}}(\tilde{\mathcal{B}})\,N_{\tilde{y}}\left(\tilde{\mathcal{B}}\right)\,\min\{\#\tilde{B}, N_B\} + \tfrac{1}{\max\{\tilde{\mathcal{B}}\} - \min\{\tilde{\mathcal{B}}\} + 1} \\ \text{s.t.} \quad & \tilde{\mathcal{B}} \subseteq \{1 - N_{PB}, \ldots, k-1\} \end{aligned} \tag{4.8}$$

where $\{1 - N_{PB}, \ldots, k-1\}$ is the only difference from Section 3.2. In this context the ensemble data driven DNN model simulated from sample N_s with data from trial k equals

$$\hat{y}_k[n] = \frac{1}{N_m} \sum_{l=1}^{N_m} \hat{y}[n \mid k, N_s, \mathcal{W}_l(\mathcal{B}_k, \max\{\mathcal{B}_k\})], \tag{4.9}$$

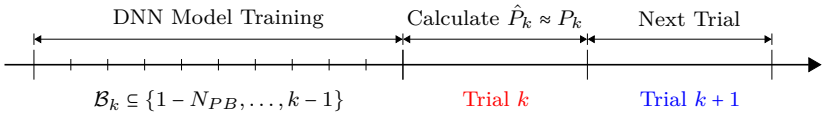


Figure 4.2: Visual segmentation of how training trials are selected and used.

where trial $\max\{\mathcal{B}_k\}$ is used for cross validation during training of the model weights $\mathcal{W}_l(\mathcal{B}_k, \max\{\mathcal{B}_k\})$. This is equivalent to evaluation forecasting in the context of Section 3.3.1 (Evaluation and Prediction Forecasting).

The super-vector ensemble data driven model $\hat{P}_k \approx P_k$ required for (4.5) is obtained by linearizing (4.9) along the trajectory of U_k using the first order Taylor expansion

$$Y_k(U) \approx \hat{Y}_k + \hat{P}_k(U - U_k) = \hat{P}_k U + \hat{K}_k \tag{4.10}$$

with

$$\hat{P}_k = \left.\frac{d\hat{Y}_k}{dU_k^T}\right|_{U_k} \text{ and } \hat{K}_k = \hat{Y}_k(U_k) - \hat{P}_k U_k$$

where $U \in \mathbb{R}^{N_u N_n}$ is the super-vector input used in (4.1a). The data driven model is retrained for every trial k. The implementation of (4.10) differs slightly from the presented as detailed in Paper D.

4.1.2 Broiler FCR Optimization using ILC

The aim is to optimize broiler growth by minimizing the FCR as discussed in Section 1.1 using the ILC algorithm presented in (4.5), which translates into using less feed to produce the same amount of broiler meat. This requires a suitable target reference R to be specified, which requires application specific knowledge about broiler growth and feed uptake behavior.

The growth of the ROSS 308 broiler strain used in this study is described by a third order polynomial by the manufacturer. The strategy is to fit a third order polynomial to feed consumption and broiler weight of the output data Y_k denoted Y_k^{fit}, which is "compressed" to emulate increased growth and "stretched" to emulate decreased feed uptake. The benefit of this approach is that it modifies the existing broiler behavior without forcing a "standard" reference, which might be unrealistic.

Broiler Weight is maximized by compressing it 1% and feed consumption is minimized by stretching it 1%. The reference is only generated at the first trial instance, $k = 1$, to investigate convergence properties of the ILC algorithm, resulting in the reference at sample n given by

$$R[n] = \begin{bmatrix} Y^{\text{fit}}_{0,\,\text{weight}}[n \cdot 1.01] \\ Y^{\text{fit}}_{0,\,\text{feed}}[n \cdot 0.99] \end{bmatrix} \tag{4.11}$$

where $Y^{\text{fit}}_{0,\text{ weight}}$ and $Y^{\text{fit}}_{0,\text{ feed}}$ is the third order weight and feed consumption polynomial.

Algorithm Configuration

The algorithm is configured identical to Section 3.2 (Dynamic Neural Network) unless specified. The DNN model output variables, $\hat{y}_k[n]$, equal: weight, cumulative feed consumption, and water consumption – *only weight and cumulative feed consumption are used by the ILC algorithm* Y_k. The DNN model input variables, $u_k[n]$, prior to IVS equal: inside temperature, humidity, outside temperature, heating rate, light intensity, CO_2, day number, and ventilation rate – *only inside temperature is used by the ILC algorithm* U_k. The algorithm uses a sample interval of 24 hours. The input variable selection algorithm is configured to find up to 2 delays among the set $\{0, 1, 2\}$. Furthermore, up to 3 significant input variables are selected from the input candidate list.

The input constraints C_l and C_u used in (4.5) are formulated as follows. The intended function of this feature is to limit the amount of input change in case of a poor model. Given that broilers are more sensitive to temperature changes early in the batch

$$c_u[n] = c_s + \frac{c_e - c_s}{N_e - N_s} \cdot (n - N_s) \text{ and} \tag{4.12a}$$

$$c_l[n] = -c_u[n], \tag{4.12b}$$

where $c_u[n]$ is an upper bound at sample n, $c_l[n]$ is a lower bound at sample n, c_s is the upper bound at sample N_s, $0 < c_s$ is the constraint at N_s while $0 < c_e$ is the constraint at sample N_e.

The input constraints for the temperature is conservatively set to N_s = day 0, N_e = day 34, c_s = 0.5 °C and c_e = 2 °C. An example of this is depicted on Figure 4.3.

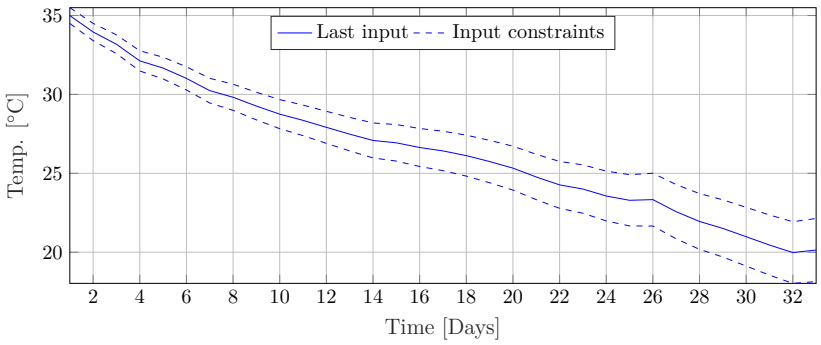


Figure 4.3: Example of temperature constraints with N_s = day 0, c_s = 0.5 °C, N_e = day 34 and c_e = 2 °C. The last input corresponds to $U_k[n]$, and the input constraints correspond to $U_k[n] + c_u[n]$ and $U_k[n] + c_l[n]$.

The norm optimal ILC weight matrices given by $W_{\Delta U}$ and W_E are selected to be constant diagonal matrices for simplicity using Bryson's rule according to

$$W_{\Delta U} = \text{diag}(W_{\Delta U}[N_s], \cdots, W_{\Delta U}[N_e]) \text{ and} \tag{4.13a}$$
$$W_E = \text{diag}(W_E[N_s], \cdots, W_E[N_e]) \tag{4.13b}$$

with

$$W_{\Delta U}[n] = (1\,^\circ\text{C})^{-2} \text{ and}$$
$$W_E[n] = \text{diag}\Big((12.5\,\text{g})^{-2}, (1.6 \cdot 12.5\,\text{g})^{-2}\Big),$$

where the constant 1.6 is a fairly common feed conversion ratio and the 12.5 g weight cost is heuristically chosen.

Normalized Performance Index

It is a long standing problem for the broiler industry to measure broiler weight accurately onwards of day 15-21 as discussed in Chapter 1 (Introduction). For this reason, the slaughter weight reported by the slaughter house is used to evaluate the performance of the algorithm, as it is an accurate sample mean produced by weighing all broilers. Note that the amount of feed is known to be accurately measured. Because the batches have different durations and slaughter weight, they are normalized by calculating the FCR at day 34, denoted FCR@34 for convenience, using

$$J_{\text{FCR},34}(y_f, y_w, t) = \frac{(y_f - k_f)\,34 \text{ days} + k_f t}{(y_w - k_w)\,34 \text{ days} + k_w t}, \tag{4.14}$$

where $y_f \in \mathbb{R}_+$ is the average feed consumed per broiler, $y_w \in \mathbb{R}_+$ is the average slaughter weight, $t \in \mathbb{R}_+$ is the slaughter age in days, and $k_w = -1.110$ kg and $k_f = -3.081$ kg are correction factors. This expression is created using official regression formulas used by the Danish broiler industry[Det Danske Fjerkæraad, 2013, pp. 85].

Algorithm Summary

Starting with $k = 1$, the Broiler ILC algorithm to calculate the next input U_{k+1} from the measured output $Y_k(U_k)$ is as follows:

1) Calculate the target reference R as described in Section 4.1.2.

2) Train N_{models} DNN models according to Section 3.2 (Dynamic Neural Network) using the algorithm configurations in Section 4.1.2.

3) Calculate the ensemble data driven system matrix $\hat{\bar{P}}_k$ in (4.5) as described in (4.10).

4) Calculate the next input U_{k+1} using the algorithm in (4.5), the constraints, target reference calculated in 1) and the weight matrices as described in Section 4.1.2.

5) Get approval from a broiler application export and the farmer, then apply U_{k+1} to batch $k+1$.

6) Record slaughter weight on completion of batch $k+1$ then $k \leftarrow k+1$ and goto step 2).

4.1.3 Experimental Results

A total of 4 trials where carried out on a state-of-the-art broiler house located in northern Denmark between September 2017 and March 2018, corresponding to almost 6 months of testing. The 10 most recent batches prior to the test are used for comparison, which are denoted pretest in the following.

The results from the house is depicted on Figure 4.4. Due to extraordinarily cold weather the heating system could not cope sufficiently in batch $k \in \{3, 4\}$. The measured temperature is depicted on the bottom left of Figure 4.4, where batch $k = 3$ has a distinct negative spike at day 10 of 5 °C, and batch $k = 4$ has a small spike of 3 °C at day 4 and an offset of 2-3 $°C$ from the reference onwards of day 20. A broiler application expert considers this difference significant, especially in the beginning of the batch where the broilers are most sensitive to temperature deviations, hence, these batches are discarded when evaluating the proposed algorithms performance.

Trial $k \in \{1, 2\}$ progressively converge towards the reference R, as intended, where $k = 2$ is very close. This convergence behavior is unrealistic, as it is expected to asymptotically converge to R – this is, however, caused by a positive weight measurement bias as discussed previously. The FCR for $k \in \{1, 3, 4\}$ are all quite close to the pretest mean.

The "stretching"-based reference generating strategy appears to be applied on a particularly high performing batch, $k = 0$, which means that the ILC algorithm favors a heavier broiler with more feed compared to the pretest mean. This is not considered an issue, however, it is unexpected that FCR is lowered by increasing the feed uptake. The measured broiler weight is high for $k \in \{1, 2\}$, while the feed consumption only increases for $k = 1$.

In Table 4.1 the normalized test results are depicted. The FCR@34 is slightly worse for batch $k = 1$ with an FCR@34 increase of 0.013, which is within a standard deviation of the pretest mean. Batch $k = 2$ has a 0.025 lower FCR@34 compared to the pretest mean, corresponding to almost 2 standard deviations, which appear very promising. Hence, the trend appears to be negative from the first two batches, exactly as desired, but further testing is required to evaluate the performance of the proposed method.

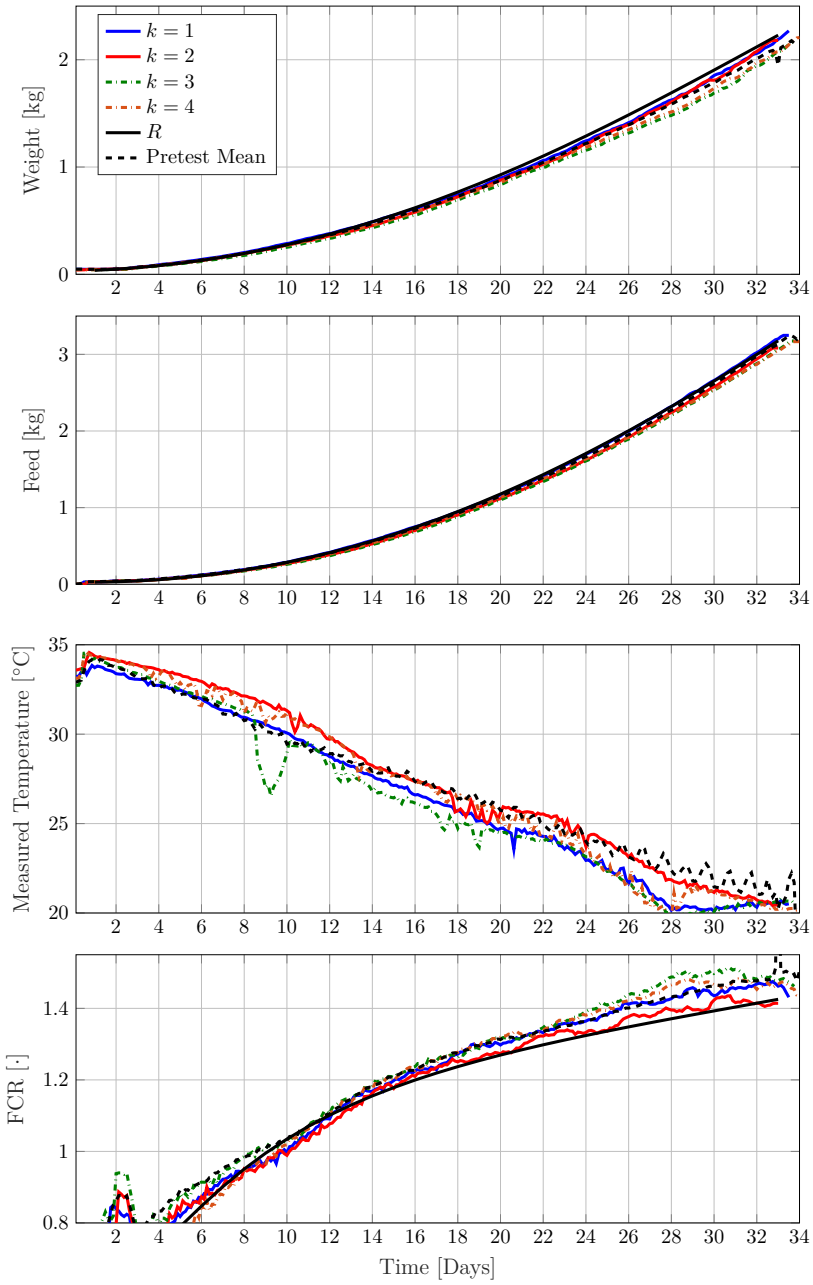


Figure 4.4: Results of applying the proposed ILC algorithm. The reference R is generated as described in Section 4.1.2.

	FCR@34 [·]	**Note**
Pretest	1.486 ± 0.017	mean ± std
R	1.445 (-0.043)	
$k = 1$	1.499 (0.013)	
$k = 2$	1.461 (-0.025)	
$k = 3$	1.544 (0.058)	Discarded
$k = 4$	1.549 (0.063)	Discarded

Table 4.1: Normalized Experimental Results. The pretest batches consists of 10 prior batches before the test, where $(\cdot)$ denotes the relative value to the pretest mean.

4.2 FCR Optimization using Terminal Iterative Learning Control

4.2.1 Heuristic Broiler Growth Model

The heuristic broiler FCR model developed in this section is used to test the data driven broiler growth optimization algorithm developed in Section 4.2.3 in a simulation environment prior to experimental tests. Only past growth model data, and not the growth model, is used for control synthesis, which would also be the case under real production conditions. The objective is to represent basic broiler growth behavior in an industrial *state-of-the-art* broiler production, which is based on the experience and knowledge of a broiler application expert.

The model's primary objective is to asses the algorithm's ability to iteratively learn a unique time series of broiler state dependent temperature inputs that minimizes the terminal broiler FCR, while simulating reduced growth for both negatively- and positively suboptimal temperature inputs. Such a broiler growth model can be represented by the discrete time dynamic nonlinear model

$$\begin{bmatrix} x_m[n+1] \\ x_f[n+1] \end{bmatrix} = \begin{bmatrix} x_m[n] \\ x_f[n] \end{bmatrix} + T_s \begin{bmatrix} G(u[n], x_m[n]) \\ R_f(x_m[n]) \end{bmatrix}, \tag{4.15a}$$

$$\begin{bmatrix} z_w[n] \\ z_f[n] \end{bmatrix} = \begin{bmatrix} R_w(x_m[n]) \\ x_f[n] \end{bmatrix} \text{ and} \tag{4.15b}$$

$$\begin{bmatrix} y_w[n] \\ y_f[n] \end{bmatrix} = \begin{bmatrix} z_w[n] \\ z_f[n] \end{bmatrix} + \begin{bmatrix} q_w[n] + q_{w,\text{bias}}[n] \\ q_f[n] \end{bmatrix} \tag{4.15c}$$

with initial conditions $x_m[N_s] = x_f[N_s] = 0$ and measured slaughter weight $\Gamma = z_w[N_e]$, where $x_m[n] \in \mathbb{R}_+$ is the broiler maturity in "effective growth days", $z_w[n] \in \mathbb{R}_+$ is the true broiler weight, $y_w[n] \in \mathbb{R}_+$ is the measured broiler weight, $x_f[n] \in \mathbb{R}_+$ is the cumulative feed consumption, $z_f[n] \in \mathbb{R}_+$ is the true cumulative feed consumption, $y_f[n] \in \mathbb{R}_+$ is the measured cumulative feed consumption, $u[n] \in \mathbb{R}$ is the temperature input, and $T_s \in \mathbb{R}_+$ is the sampling interval in days. Under production conditions the temperature input $u[n]$ is a reference for the climate control system, which for simplicity is assumed

to achieve perfect tracking. In (4.15a), the smooth maturation rate function $G{:}\,\mathbb{R}\times\mathbb{R}_+ \to [\beta, 1]$ where $\beta \in [0, 1[$ is the worst case broiler growth rate. Finally, $R_w{:}\,\mathbb{R}_+ \to \mathbb{R}_+$ and $R_f{:}\,\mathbb{R}_+ \to \mathbb{R}_+$ are smooth and strictly increasing functions mapping the broiler maturity $x_m[n]$ into broiler weight and feed consumption, $q_w[n] \in \mathbb{R}$ is the weight measurement noise, $q_{w,\text{bias}}[n] \in \mathbb{R}$ is the weight bias, and $q_f[n] \in \mathbb{R}$ is the feed measurement noise.

The growth and feed consumption of the widely used ROSS 308 fast growing broiler strain are described by the manufacturer in [Aviagen, 2014a, pp. 3] by

$$R_w(t) = \frac{-18.3\,t^3 + 2.2551\,t^2 + 2.9118\,t + 54.739}{1000} \text{ and} \tag{4.16a}$$

$$R_f(t) = \frac{21.9\cdot 10^{-6}\,t^4 - 4.232\cdot 10^{-3}\,t^3 + 0.206\,t^2 + 2.02\,t + 11.6}{1000}, \tag{4.16b}$$

where $R_w(t) \in \mathbb{R}_+$ is the broiler weight reference in kg, $R_f(t) \in \mathbb{R}_+$ is the broiler feed uptake reference in kg/day, and $t \in [0, 59]$ days is the time in "effective growth days". Expressing broiler weight $R_w(x_m[n])$ and broiler feed uptake $R_f(x_m[n])$ in terms of the broiler maturity in "effective growth days" through $x_m[n]$ results in realistic weight and feed uptake behavior, as it accurately captures the nonlinear nature of broiler growth.

A modified normal distribution is an appropriate representation for the maturation rate function G, as it has a unique maximum, and the standard deviation can easily be tuned to design how sensitive G is to temperature errors. Specifically

$$G(u[n], x_m[n]) = \beta + (1-\beta)\exp\left\{\ln\left(\frac{\alpha+\beta-1}{\beta-1}\right)\left[\frac{u[n]-\bar{u}(x_m[n])}{\sigma_u}\right]^2\right\}, \tag{4.17}$$

where $\bar{u}(x_m[n])$ is the temperature maximizing G, $G(\bar{u}(x_m[n]), x_m[n]) = 1$, and $\sigma_u \in \mathbb{R}_+$ is the constant temperature sensitivity. The temperature sensitivity is the temperature input error, $u[n] - \bar{u}(x_m[n])$, resulting in a decreased maturation rate of α – corresponding to $G(\bar{u}(x_m[n]) \pm \sigma_u, x_m[n]) = 1-\alpha$ with $\alpha \in]0, 1-\beta[$. The intention is to allow the broiler application expert to heuristically specify the decreased growth rate for a specific temperature error. In Figure 4.5, the components of the maturation rate function G are shown.

The optimal temperature profile is unknown in the industry, but typical temperature profiles for the ROSS 308 fast growing broiler transition almost linearly between the initial temperature of $\bar{u}_s = 34$ °C at day $t_s = 0$ to $\bar{u}_e = 21$ °C at day $t_e = 34$. This corresponds to a temperature drop of $(\bar{u}_e - \bar{u}_s)$, which is modeled as proportional to the maturity $x_m[n]$ as

$$\bar{u}(x_m[n]) = \bar{u}_s + \Delta T x_m[n] \text{ with } \Delta T = \frac{\bar{u}_e - \bar{u}_s}{t_e - t_s}. \tag{4.18}$$

Consequently, the optimal temperature at sample n depends on $x_m[n-1]$, which, in turn, depends on all prior inputs.

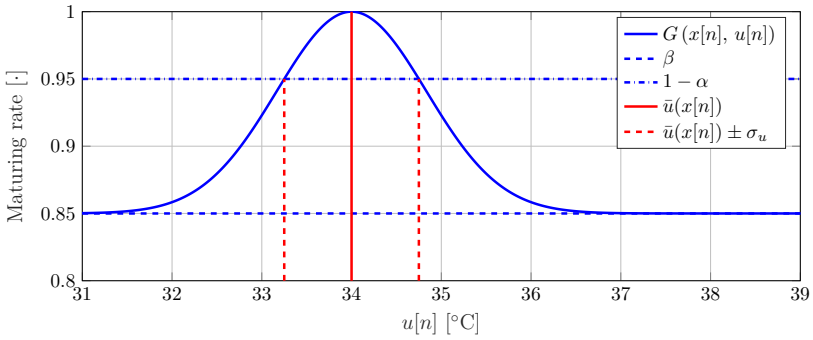


Figure 4.5: Visualization of the maturation rate function $G(x_m[n], u[n])$ for $x_m[n] = 0$ with worst case broiler growth rate $\beta = 0.85$, $\alpha = 0.05$, maximizing input $\bar{u}(x_m[n]) = 34\,[°\mathrm{C}]$ and temperature error sensitivity $\sigma_u = 0.75\,[°\mathrm{C}]$.

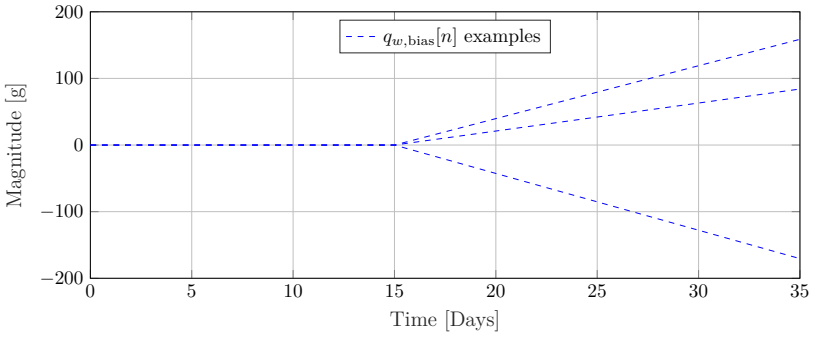
The weight bias term $q_{w,\mathrm{bias}}[n]$ was investigated in Section 3.4 (Improved Slaughter Weight Forecasting) and found to cause terminal weight measurement errors, $\tilde{y}_w - \tilde{z}_w$, with −27.4g mean and 115.9g standard deviation through comparison with the accurately measured slaughter weight. The weight bias onset was found to occur around day 15, which is heuristically assumed to increase linearly from zero at day 15 to $\mathcal{Q}_{\mathrm{bias}} \sim \mathcal{N}(-27.4\mathrm{g}, 115.9\mathrm{g})$ at the terminal sample and hence

$$z_{w,\mathrm{bias}}[n] = \begin{cases} \frac{nT_s - 15}{N_e T_s - 15}\mathcal{Q}_{\mathrm{bias}}, & 15 < nT_s \\ 0, & \text{otherwise} \end{cases}, \tag{4.19}$$

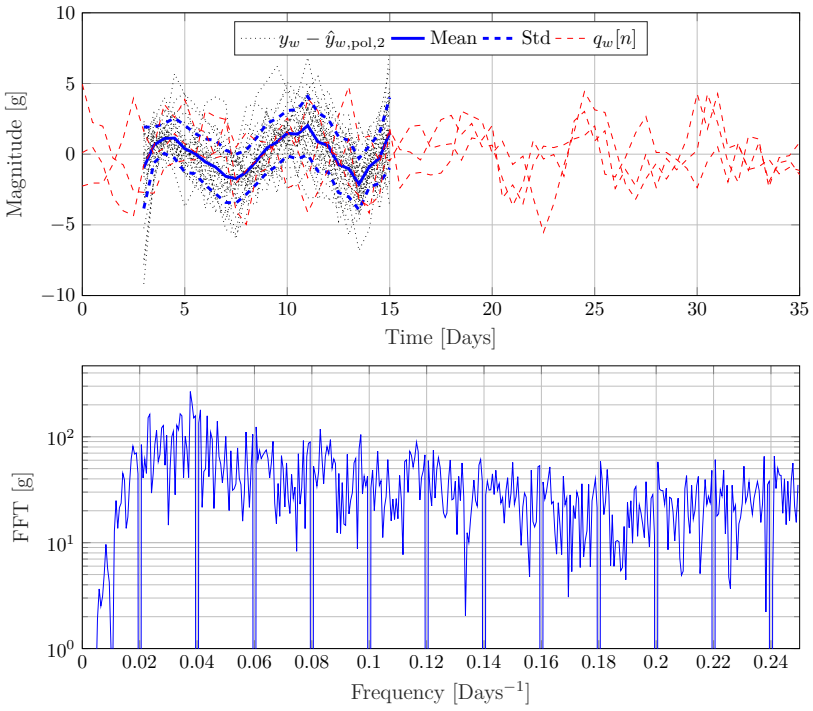
where $\mathcal{Q}_{\mathrm{bias}}$ is constant throughout each simulation as shown in Figure 4.6a. In Section 3.4 it was found that using the measured slaughter weight, i.e. the terminal broiler weight, reduces the weight bias effect for broiler weight prediction on real broiler production data.

The noise terms $q_w[n]$ and $q_f[n]$ are found by analyzing the frequency spectrum of production data from the experimental test site. As broiler weight is a smooth function of time, the "true" broiler weight is approximated by a second order polynomial $\hat{y}_{w,\mathrm{pol},2}$ between day 3 and 15, where the weight measurement y_w is expected to be the most reliable. The fit errors, $y_w - \hat{y}_{w,\mathrm{pol},2}$, of 36 batches from the experimental test site are shown in the top plot of Figure 4.6b and is assumed to be measurement noise.

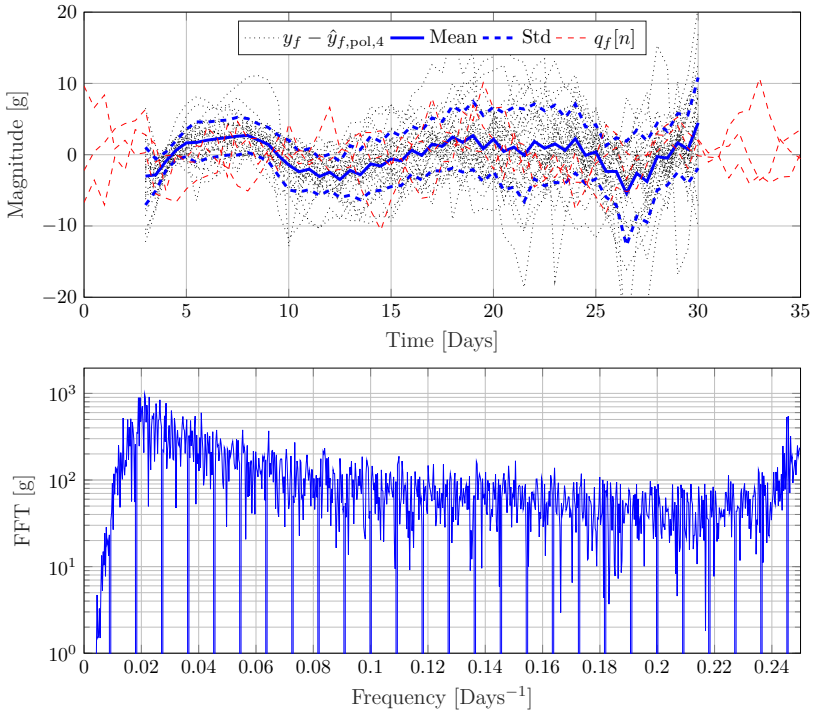
Subtracting the mean, concatenating all the fit errors and computing the FFT produces the bottom magnitude plot. As this is not a standard distribution, random realizations of $q_w[n]$ with identical magnitude are obtained by randomly rotating the phases of the FFT and applying the inverse discrete Fourier transform. For more information on this approach, see [Prichard and Theiler, 1994]. Some realizations of $q_w[n]$ are shown in the top plot of Figure 4.6b. Similarly, the "true" cumulative feed uptake is approximated by a fourth order polynomial $\hat{y}_{f,\mathrm{pol},4}$ between day 3 and 30 and shown in Figure 4.6c.



(a) Weight measurement bias $q_{w,\text{bias}}[n]$ samples using (4.19).



(b) Visualization of the weight measurement noise $q_w[n]$.



(c) Visualization of the feed uptake measurement noise $q_f[n]$.

Figure 4.6: Measurement behavior for the heuristic broiler growth model.

Control Design Considerations

In this section optimization strategies of the heuristic broiler growth model are discussed. The heuristic model can explain the rationale between two of the popular strategies used in *state-of-the-art* broiler production that is considered next. The first of these maintains a higher than necessary temperature to dry out the litter to limit ammonia development and subsequently sickness. The second initially reduce maturation through special light programs to properly develop the broiler, followed by unrestricted growth period for the remaining duration of the batch.

Weight maximization considerations Inspecting G shows that $x_m[n]$ is maximized by the unique input $\bar{u}(x_m[n])$ that for all $u[n] \neq \bar{u}(x_m[n])$ satisfies

$$G(u[n], x_m[n]) < G(\bar{u}(x_m[n]), x_m[n]) = 1.$$

In the case when $\beta \leq G \leq 1$, the largest possible maturity $\bar{x}_m[n]$ equals

$$\bar{x}_m[n] = \max\{x_m[n]\} = T_s \sum_{i=1}^{n} \max\{G(u[i], x_m[i])\} = n\,T_s.$$

As R_w is strictly increasing, the largest possible true broiler weight $\bar{z}[n]$ is given by

$$\bar{z}_w[n] = \max\{z_w[n]\} = \max\{R_w(x_m[n])\} = R_w(\max\{x_m[n]\}) = R_w(n\,T_s).$$

This ensures that suboptimal control results in suboptimal growth, as expected in real broiler production where either a too low or too high temperature results in decreased broiler growth.

In Figure 4.7 the behavior of the broiler model is shown for different temperature inputs. Interestingly, if the temperature is positively suboptimal, e.g., $u[n] = \bar{u}(\bar{x}_m[n]) + 1\,^\circ\mathrm{C}$, then G converges to 1 for $n \to \infty$, and if the temperature is negatively suboptimal, e.g., $u[n] = \bar{u}(\bar{x}_m[n]) - 1\,^\circ\mathrm{C}$, then G converges to β for $n \to \infty$. This is caused by the decreasing $\bar{u}(x_m[n])$ for increasing $x_m[n]$ – supported by the fact that broiler farmers tend to use positively suboptimal temperatures, similar to the first *state-of-the-art* strategy.

Feed minimization considerations If $\beta \leq G \leq 1$, the smallest maturation rate $\underline{x}_m[n]$ is governed by

$$\begin{aligned} \underline{x}_m[n] &= \min\{x_m[n]\} = T_s \sum_{i=1}^{n} \min\{G(u[i], x_m[i])\} \\ &= T_s \beta n \end{aligned} \tag{4.20}$$

As R_f is strictly increasing, the lowest cumulative feed consumption is given by

$$\begin{aligned}\underline{x}_f[n] &= \min\{x_f[n]\} = \min\left\{T_s \sum_{i=1}^{n} R_f(x_m[i])\right\} \\ &= T_s \sum_{i=1}^{n} R_f(\min\{x_m[i]\}) = T_s \sum_{i=1}^{n} R_f(T_s \beta i)\end{aligned} \tag{4.21}$$

This suggests that feed minimization and weight maximization are completely opposing goals.

FCR minimization considerations The expression for FCR from the heuristic model is

$$z_{\text{FCR}}[n] = \frac{z_f[n]}{z_w[n]} = \frac{x_f[n]}{R_w(x_m[n])} = T_s \frac{\sum_{i=1}^{n} R_f(x_m[i])}{R_w(x_m[n])} \tag{4.22}$$

Unlike weight maximization and feed minimization, an analytical expression for the lowest possible FCR is non-trivial to determine, as depicted on

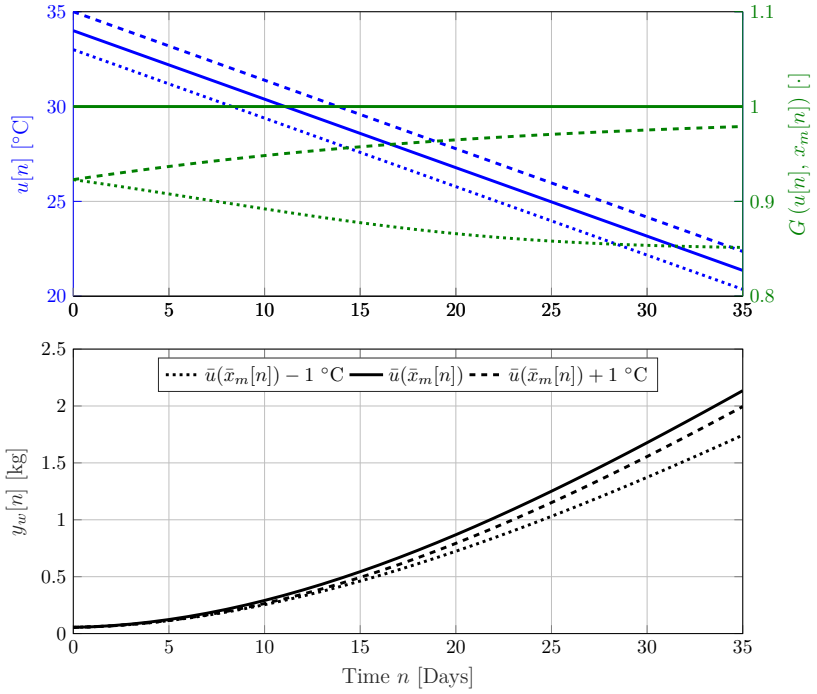


Figure 4.7: Visualization of broiler growth y_m with different inputs. The top plot depicts the maturation rate function $G(u[n], x_m[n])$ as a function of the input $u[n]$ and the bottom plot depicts the output $y_m[n]$. The model settings equal that of Figure 4.5 with $T_s = 1$ day.

Figure 4.8. This is due to two simultaneous and opposing objectives, namely weight maximization and feed minimization, which turns out to depend on the simulation duration N_e as depicted on Figure 4.8. In Figure 4.9 the proposed strategies are compared, from which it follows that FCR minimization consists of an initial period of feed minimization followed by weight maximization – similar to the second *state-of-the-art* strategy. *Feed minimization* produces the highest FCR, and is therefore ruled out. Moreover, *weight maximization* results in a 1.1% higher FCR than *FCR minimization*, which makes *FCR minimization* favorable despite the added complexity of another output, and will therefore be used in this work.

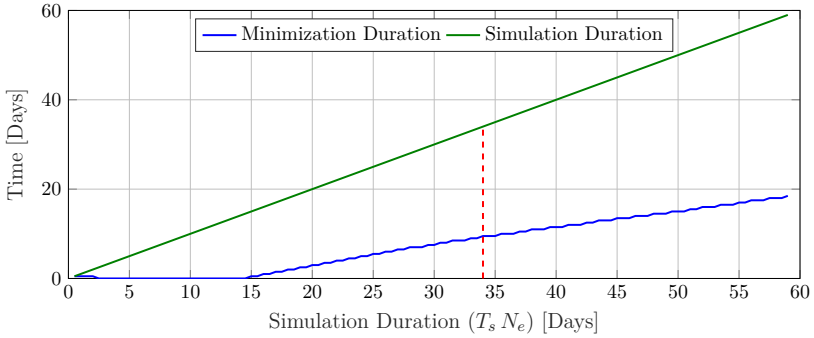


Figure 4.8: Minimization duration as a function of simulation duration ($N_e T_s$) with β = 0.85 and T_s = 0.5 days, which corresponds to the length of the initial period where $G(u[n], x_m[n]) = \beta$. The red dashed line indicates a simulation duration of 34 days with a minimization duration of 9.5 days, equivalent to Figure 4.9.

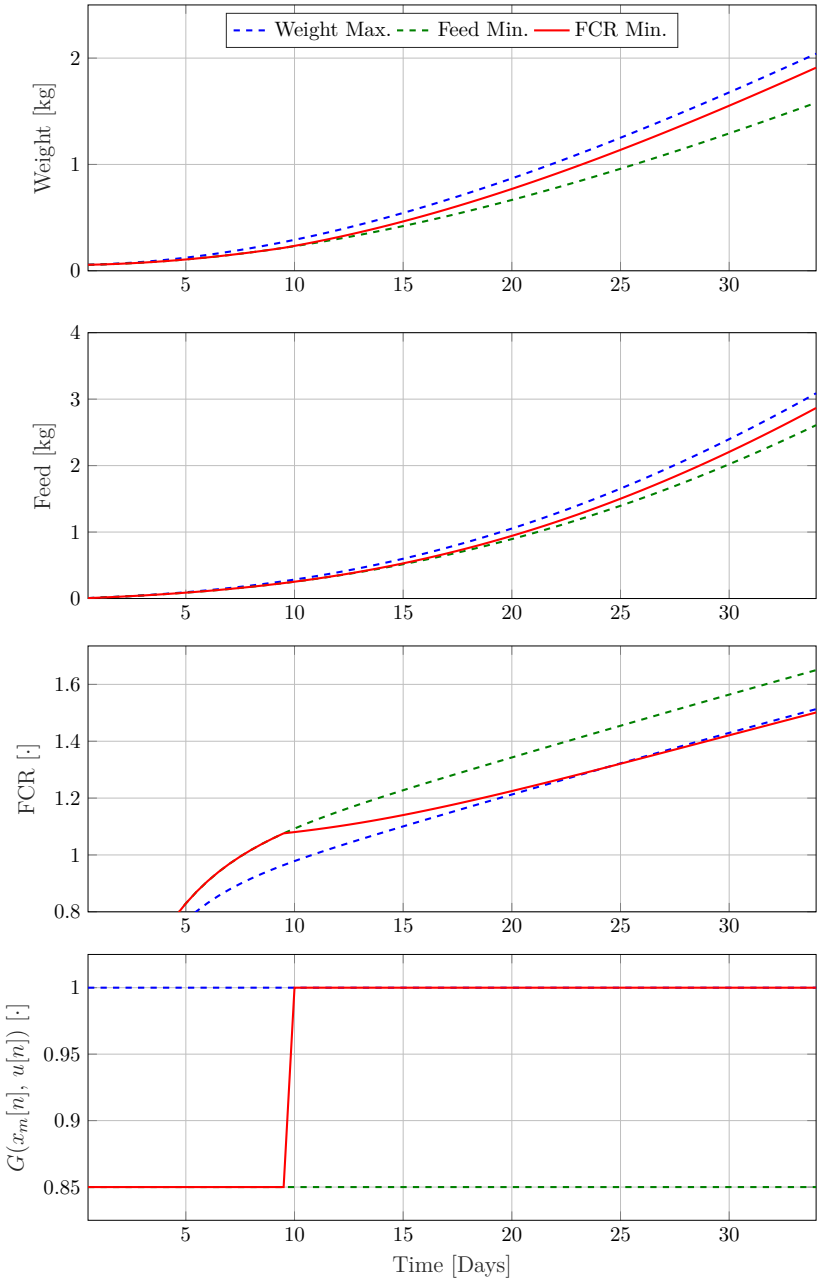


Figure 4.9: Visualization of different optimization strategies with $N_e = 34$ days, $T_s = 1$ day and $\beta = 0.85$. A FCR difference of $1.14 \cdot 10^{-3}$, equivalent of 1.1%, exists between *Growth maximization* and *FCR minimization*, which potentially makes *FCR minimization* a better strategy.

4.2.2 Terminal Iterative Learning Control

Terminal ILC (TILC) is a method that can be applied to a repeating process with the aim of iteratively learning the input sequence $U_k \in \mathbb{R}^{N_u N_n}$ such that the terminal process output $\tilde{Y}_k(U_k) = y_k[N_e] \in \mathbb{R}^{N_y}$ tracks the desired terminal reference $\tilde{R} = r[N_e] \in \mathbb{R}^{N_y}$ denoted by

$$\lim_{k\to\infty} \tilde{Y}_k(U_k) = \tilde{R}, \tag{4.23}$$

with the super-vector model used for control synthesis given by

$$\tilde{Y}_k(U_k) = \tilde{P}U_k + \tilde{K}, \tag{4.24}$$

where $\tilde{P} \in \mathbb{R}^{N_y \times N_u N_n}$ is the terminal system matrix, and $\tilde{K} \in \mathbb{R}^{N_y}$ represents terminal effects unrelated to the input U_k. This is almost equivalent to Section 4.1.1 (Iterative Learning Control) with the exception of the terminal output notation, $\tilde{\cdot}$.

This problem can be solved using constrained Norm Optimal Point-To-Point ILC, which aims at tracking the output at specific samples. TILC is a specialization of Point-To-Point ILC, as TILC only aims at tracking the terminal output. Adapting the constrained Norm Optimal Point-To-Point ILC algorithm 1 in [Chu et al., 2015] to the special case of the TILC problem gives

$$U_{k+1} = \arg\min_{U\in\Omega} \|\tilde{E}_k(U)\|^2_{W_{\tilde{E}}} + \|U - U_k\|^2_{W_{\Delta U}} \tag{4.25a}$$

subject to

$$\tilde{E}_k(U) = \tilde{R} - \tilde{Y}_k(U) \text{ and} \tag{4.25b}$$

$$\tilde{Y}_k(U) = \tilde{P}U + \tilde{K}, \tag{4.25c}$$

where Ω is the set of valid inputs, $W_{\tilde{E}} \in \mathbb{R}^{N_y \times N_y}$ is the symmetric positive definite tracking error cost matrix, $W_{\Delta U} \in \mathbb{R}^{N_u N_n \times N_u N_n}$ is the symmetric positive definite input change cost matrix and $\tilde{E}_k(U) \in \mathbb{R}^{N_y}$ is the terminal tracking error given by (4.25b). The intuition behind (4.25) is to reduce the terminal tracking error, e.g at slaughter, by finding an input in the neighborhood of U_k that minimizes the cost function (4.25a).

The following results were established in [Chu et al., 2015].

Theorem 1. *If perfect tracking is feasible, i.e.* $\exists U \in \Omega$ *such that* $\tilde{Y}_k(U) = \tilde{R}$*; then (4.25) achieves monotonic convergence to zero tracking error*

$$\left\|\tilde{E}_{k+1}(U_{k+1})\right\|_{W_{\tilde{E}}} \leq \left\|\tilde{E}_k(U_k)\right\|_{W_{\tilde{E}}} \quad \forall k \in \mathbb{Z}_+ \tag{4.26}$$

and

$$\lim_{k\to\infty} \tilde{E}_k(U_k) = 0, \lim_{k\to\infty} U_k = \bar{U}. \tag{4.27}$$

Theorem 2. *If perfect tracking is not feasible, i.e.* $\tilde{Y}_k(U) \neq \tilde{R} \quad \forall U \in \Omega$*; then the input of (4.25) converges to*

$$\lim_{k\to\infty} U_{k+1} = \arg\min_{U\in\Omega} \|\tilde{R} - \tilde{P}U - \tilde{K}\|^2_{W_{\tilde{E}}}, \tag{4.28}$$

equivalent of the algorithm converging to the smallest possible tracking error. Moreover, this convergence is monotonic in the tracking error norm

$$\left\|\tilde{E}_{k+1}(U_{k+1})\right\|_{W_{\tilde{E}}} \leq \left\|\tilde{E}_k(U_k)\right\|_{W_{\tilde{E}}} \quad \forall k \in \mathbb{Z}_+. \tag{4.29}$$

4.2.3 Data Driven TILC Broiler FCR Minimization

Super-vector Data Driven TILC Broiler FCR Model

FCR minimization requires an augmented DNN FCR model, denoted by $(\cdot)^*$. This model is given by

$$\tilde{Y}^*_k(U) = \frac{\tilde{Y}_{k,f}(U)}{\tilde{Y}_{k,w}(U)} \tag{4.30}$$

where $\tilde{Y}_{k,w}(U) \in \mathbb{R}_+$ and $\tilde{Y}_{k,f}(U) \in \mathbb{R}_+$ respectively denote the weight and cumulative feed uptake.

The data driven DNN model is different from Section 4.1 (FCR Optimization using Iterative Learning Control), as it uses the broiler weight bias compensation investigated in Section 3.4 (Improved Slaughter Weight Forecasting), i.e., $\mathcal{W}^*(\cdot)$ instead of $\mathcal{W}(\cdot)$. Hence, the ensemble data driven DNN model simulated from sample N_s with data from trial k equals

$$\hat{y}^*_k[n] = \frac{1}{N_m}\sum_{l=1}^{N_m} \hat{y}[n \mid k, N_s, \mathcal{W}^*_l(\mathcal{B}_k, \min\{\mathcal{B}_k\}, N_\phi, \gamma)], \tag{4.31}$$

where trial $\min\{\mathcal{B}_k\}$ is used for cross validation during training of the model weights $\mathcal{W}^*_l(\mathcal{B}_k, \min\{\mathcal{B}_k\}, N_\phi, \gamma)$ instead of $\max\{\mathcal{B}_k\}$ as in Section 4.1. Linearizing along the trajectory of U_k of (4.30) by a first order Taylor expression, similar to (4.10), results in

$$\tilde{Y}^*_k(U) \approx \hat{\tilde{Y}}^*_k(U_k) + \hat{\tilde{P}}^*_k(U - U_k) = \hat{\tilde{P}}^*_k U + K^*_k \tag{4.32}$$

with

$$\hat{\tilde{P}}^*_k = \frac{d\hat{\tilde{Y}}^*_k(U)}{d\hat{\tilde{Y}}^T_k(U)} \frac{d\hat{\tilde{Y}}_k(U)}{dU^T} = \frac{d\hat{\tilde{Y}}^*_k(U)}{d\hat{\tilde{Y}}^T_k(U)} \hat{\tilde{P}}_k \text{ and}$$

$$\hat{\tilde{K}}^*_k = \hat{\tilde{Y}}^*_k(U_k) - \hat{\tilde{P}}^*_k U_k = \frac{d\hat{\tilde{Y}}^*_k(U)}{d\hat{\tilde{Y}}^T_k(U)} \hat{\tilde{K}}_k.$$

Data Driven TILC Broiler FCR Minimization

The objective is to minimize terminal broiler FCR, which is unknown in broiler production. One reason for this is that artificial genetic selection progressively increases the growth rate. To account for this, the reference is redefined to

$$\tilde{R}_k^* = \tilde{Y}_k^*(U_k) - \mathcal{R}, \tag{4.33}$$

where $\mathcal{R} \in \mathbb{R}_+^{N_y}$ is a trial-independent minimization vector with positive elements. This method is termed *minimizing reference*. As $\tilde{E}_k^*(U_k) = \tilde{R}_k^* - \tilde{Y}_k^*(U_k) = -\mathcal{R}$ is constant, zero tracking error is not possible by construction. Assuming that $\tilde{Y}_k^*(U_k)$ is lower bounded by $\tilde{Y}_{\min}^* \in \mathbb{R}^{N_y}$ and in combination with Theorem 2, the aim is to achieve

$$\lim_{k\to\infty} \tilde{Y}_k^*(U_k) = \tilde{Y}_{\min}^* \text{ and } \lim_{k\to\infty} \tilde{R}_k^* = \tilde{Y}_{\min}^* - \mathcal{R}. \tag{4.34}$$

Since broiler growth is a nonlinear process, a local minimum could be obtained instead of $\tilde{Y}_{\min}^*$.

In the following the so-called best recent trial index κ_k is required and for $\tilde{Y}_i^*(U_i) \in \mathbb{R}_+$. It is defined by

$$\kappa_k = \underset{i\in[\min(k-N_B,0),k]}{\arg\min} \|\tilde{Y}_i^*(U_i)\|_{W_{\tilde{E}}}, \tag{4.35}$$

and serves as a feasible substitute for the *true best recent trial index* given by

$$\underset{i\in[\min(k-N_B,0),k]}{\arg\min} \|\tilde{Y}_{\min}^* - \tilde{Y}_i^*(U_i)\|_{W_{\tilde{E}}}.$$

The variable i is lower bounded by 0, which equals the most recent preliminary trial. Note that (4.35) is application-dependent. To reduce the influence of the measurement weight bias on κ_k, the slaughter weight Γ_k and measured cumulative feed consumption $\tilde{Y}_{k,f}(U_k)$ is used:

$$\kappa_k = \underset{i\in[\min(k-N_B,0),k]}{\arg\min} \left\| \frac{\tilde{Y}_{i,f}(U_i)}{\Gamma_i} \right\|_{W_{\tilde{E}}} \tag{4.36}$$

To account for the uncertain nature of the augmented data driven model given by (4.32), the TILC algorithm is modified into a descent type algorithm, denoted *anchoring*, by solving

$$U_{k+1} = \underset{U\in\Omega_{k+1}}{\arg\min} \left\|\tilde{E}_{\kappa_k}^*(U)\right\|_{W_{\tilde{E}}}^2 + \|U - U_{\kappa_k}\|_{W_{\Delta U}}^2 \tag{4.37a}$$

subject to (4.33), (4.36) and

$$\tilde{E}_{\kappa_k}^*(U) = \tilde{R}_{\kappa_k}^* - \tilde{Y}_{\kappa_k}^*(U) \text{ and} \tag{4.37b}$$

$$\tilde{Y}_{\kappa_k}^*(U) = \hat{\tilde{P}}_{\kappa_k}^* U + \hat{\tilde{K}}_{\kappa_k}^* \tag{4.37c}$$

where $\Omega_{k+1} \in \mathbb{R}^{N_u N_n}$ is the set of valid trial dependent inputs.

If the input U_{k+1} does not decrease the error in (4.36), then the data driven model is not sufficiently accurate in the neighborhood of U_{κ_k}, i.e. $\hat{\tilde{P}}^*_{\kappa_k} \notin \tilde{P}^*_k$. Therefore U_{k+1} is rejected and U_{k+2} is calculated in the neighborhood of $U_{\kappa_{k+1}} = U_{\kappa_k}$ instead of U_{k+1}. This effectively ensures that the algorithm keeps exploring the neighborhood of the recent best trial input U_{κ_k} until the data driven model is sufficiently accurate to maximize the terminal output norm in (4.36), as the data driven model always uses the most recent data from the last N_B trials. Consequently, if the data driven model $\hat{\tilde{P}}^*_{\kappa_k}$ is identical to the analytical model $\tilde{P}^*_{\kappa_k}$ under ideal conditions and constant reference. In this case $\kappa_k = k$ as $\tilde{E}^*_k$ is monotonically decreasing in k.

The computable solution of (4.37) has almost the same solution as (4.5) in Section 4.1.1 (Iterative Learning Control) and results in an algorithm of the form $U_{k+1} = F\big(U_{\kappa_k}, \tilde{E}^*_{\kappa_k}(U_{\kappa_k})\big) = F\big(U_{\kappa_k}, \tilde{R}^*_{\kappa_k} - \tilde{Y}^*_{\kappa_k}(U_{\kappa_k})\big)$ that includes feedback action though the measured terminal output via the terms $\tilde{Y}^*_{\kappa_k}(U_{\kappa_k})$ and $\hat{\tilde{P}}^*_{\kappa_k}$. The slaughter weight is used to calculate $\tilde{E}^*_{\kappa_k}(U_{\kappa_k})$, similar to (4.36), to reduce the influence of the weight measurement bias. If combined with maximizing reference then $\tilde{E}^*_k(U_{\kappa_k}) = \mathcal{R}$ and $\tilde{Y}^*_{\kappa_k}(U_{\kappa_k})$ is only used indirectly through $\hat{\tilde{P}}^*_{\kappa_k}$.

Analytical Heuristic Model

To evaluate the ILC algorithm formulated in Section 4.2.3 in simulation, an analytical linear terminal super-vector broiler growth model of $\tilde{Z}_k \in \mathbb{R}^{N_y}$ is required. This is obtained by linearizing (4.15) along the trajectory of $U_k \in \mathbb{R}^{N_u N_n}$ using the first order Taylor expansion by

$$\tilde{Z}_k(U) \approx \tilde{Z}_k(U_k) + \tilde{P}^*_k\,(U - U_k) = \tilde{P}^*_k U + \tilde{K}^*_k \tag{4.38}$$

with

$$\tilde{P}^*_k = \left.\frac{d\,\tilde{Z}_k(U_k)}{d\,U_k^T}\right|_{U_k} \text{ and } \tilde{K}^*_k = \tilde{Z}_k(U_k) - \tilde{P}^*_k U_k,$$

where $\tilde{P}^*_k \in \mathbb{R}^{N_y \times N_u N_n}$ is the terminal model matrix and $\tilde{K}^* \in \mathbb{R}^{N_y}$ is the terminal output constant vector unrelated to the input $U \in \mathbb{R}^{N_u N_n}$.

4.2.4 Simulation Case Study

The objective is to investigate the ability of different configurations of the data driven optimization algorithm (4.37) to minimize the terminal FCR $\tilde{Y}^*_k$ of the heuristic broiler growth model given by (4.15) using the last N_B trials, hence, $\mathcal{B}_k = \{k-N_B+1, \ldots, k-1\}$. Specifically, the performance impact of the following is investigated:

1. using the data driven model $\hat{\tilde{P}}^*_{\kappa_k}$ for control synthesis from (4.32), denoted by (D), compared to the unrealistic option of using the analytical super-vector model $\tilde{P}^*_{\kappa_k}$ for control synthesis from (4.38), denoted by (I), as shown in Figure 4.10b.

2. using anchoring from (4.37) though κ_k from (4.36), denoted by (A), compared to disabling this term by forcing $\kappa_k = k$, denoted by $(\cdot)$, as shown on Figure 4.10c.

3. using the maximizing reference (4.33), denoted by (MR), compared to the unrealistic option of using the true analytical maximum given by

$$\tilde{R}^*_k = \tilde{Y}^*_{\min} = z[N_e], \tag{4.39}$$

indicated by $(\cdot)$, as depicted on Figure 4.10d.

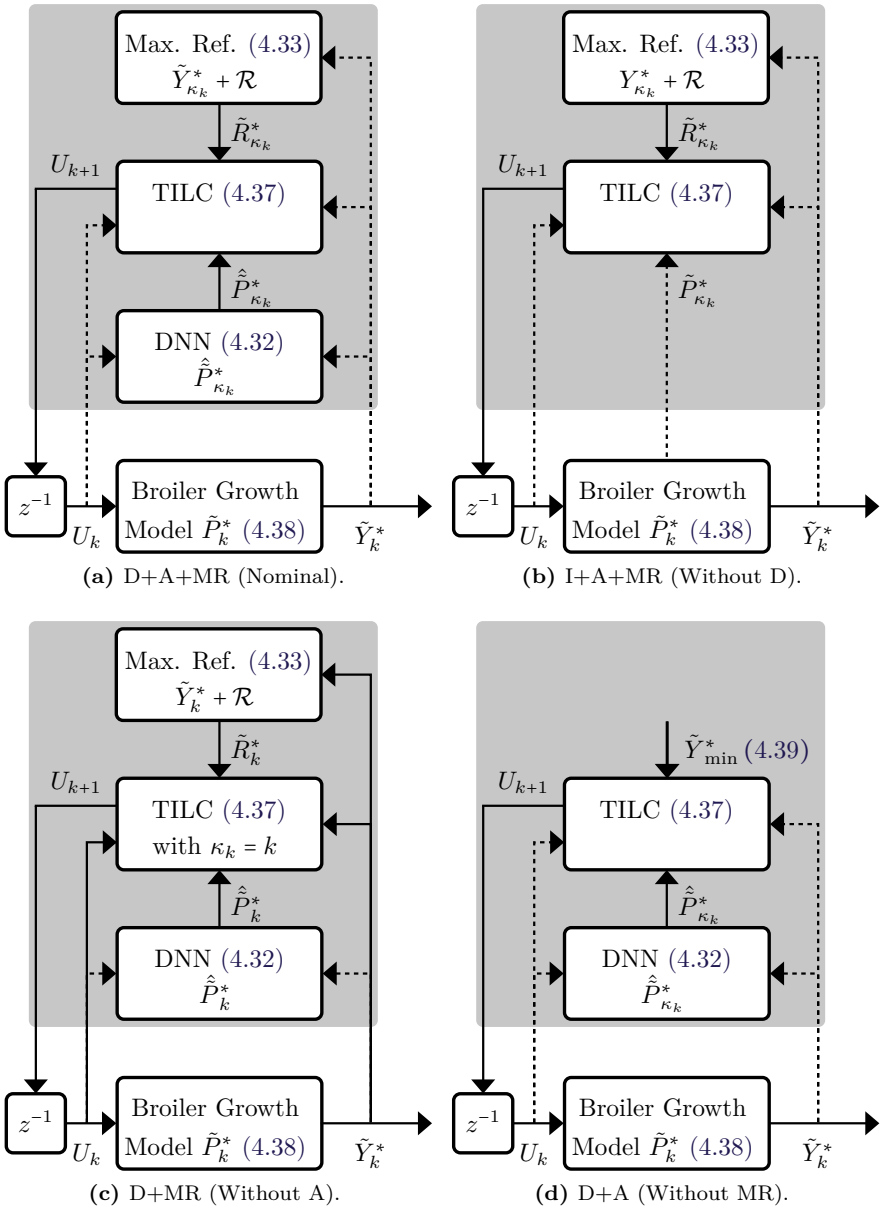
This results in a total of 8 different test configurations, some of which are shown in Figure 4.10. Each test is repeated 10 times and the mean true terminal error, $|\tilde{Z}_k - \tilde{R}_{\max}|$, is used for evaluation.

To investigate the necessity for iterative learning in this data driven application, different values of $W_{\Delta U}$ are explored under unconstrained conditions, i.e., $\Omega_k = \mathbb{R}^{N_e N_u}$. Note that by using $W_{\Delta U} = 0$ with a perfect model under linear conditions results in instantaneous convergence in a single trial [sung Ahn et al., 2006a]. Specifically, if $W_{\Delta U} = 0$ has instantaneous convergence with the D+A+MR algorithm compared to using $W_{\Delta U} > 0$, then there is no need for iterative learning.

Method and Model Configuration

The heuristic broiler growth model in Section 4.2.1 was simulated between the initial sample $N_s = 0$ and the terminal sample $N_e =$ day 35 with a sample interval of $T_s = 0.5$ days, and is heuristically configured as follows: $\beta = 0.85$ as the worst case maturing rate, since feed and water consumption are the dominating factors and correct temperature control is regarded as a catalyst. Also $\alpha = 0.05$ and $\sigma_u = 0.75\,[^\circ\text{C}]$ have been used to give good overall sensitivity throughout the lifespan of a broiler.

The data driven model in (4.32) is configured as the default algorithm in Section 3.4 (Improved Slaughter Weight Forecasting) unless specified. It is generated with $N_m = 20$ ensemble models using $N_B = 10$ training batches, $N_l = 3$ input and output lags. The required preliminary $N_{PB} = N_B$ trials at trial $t = 1$ for training are generated using the positive input $u[n]$ resulting in a 5% decreased maturing rate, $G(u[n], x_m[n]) = 0.95$, see the example in Figure 4.11. To ensure an identical initial input U_0 for all the tests, the most recent preliminary trial $k = 0$ does not have any added input noise. Hence, the objective is to decrease the terminal broiler FCR $\tilde{Y}^\star_k$ by 0.0537. White noise with standard deviation of 0.3 °C is added to the remaining $N_B - 1$ preliminary



(a) D+A+MR (Nominal). **(b)** I+A+MR (Without D).

(c) D+MR (Without A). **(d)** D+A (Without MR).

Figure 4.10: Illustration of some of the configurations of the broiler growth optimization algorithm tested in Section 4.2.4. The shaded area denotes the controller, z^{-1} denotes a unit delay, a dashed signal contains information from the last N_B trials, $\{k - N_B + 1, \ldots, k\}$, and a non-dashed signal only contains information from trial k. See Section 4.2.4 for detailed explanation.

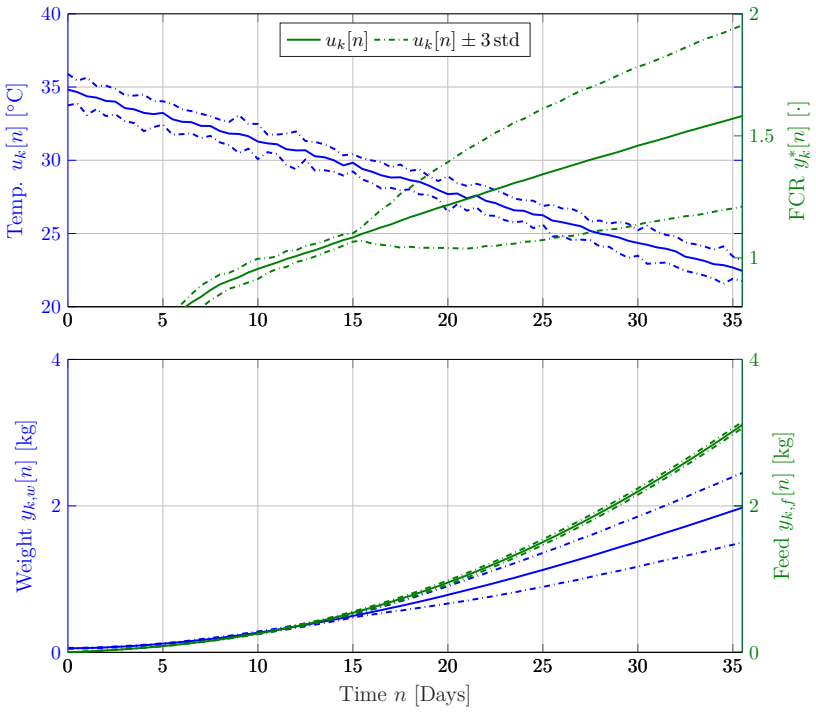


Figure 4.11: Visualization of 10 preliminary trial data. Note that the large FCR standard deviation is caused by by the measured weight bias $q_{w,\mathrm{bias}}[n]$.

trials, $\{1 - N_B, \ldots, -1\}$. This is considered realistic, as most broiler farmers tend to use a too high temperature with little variations from trial-to-trial.

Fast convergence conditions for the data driven TILC broiler optimization algorithm are obtained by using a minimization constant of $\mathcal{R} = 0.04$, terminal tracking error cost and input change cost of $W_{\tilde{E}} = 0.01^{-2}$ and $W_{\Delta U} = \mathrm{diag}\big([1\,^\circ\mathrm{C}]^{-2}, \ldots, [1\,^\circ\mathrm{C}]^{-2}\big)$. The permitted temperature change is restricted to avoid large input fluctuations caused by data driven modeling errors in $\hat{\hat{P}}^*_{\kappa_k}$. The valid input space Ω_{k+1} is therefore given by:

$$\begin{aligned} \omega_{k+1}[n] &= \{u \,|\, -\gamma[n] \leq u - u_{\kappa_k}[n] \leq \gamma[n]\} \text{ with} \\ \gamma[n] &= 0.5\,^\circ\mathrm{C} + n\,T_s\,\frac{1.5\,^\circ\mathrm{C}}{35\,\mathrm{Days}} \end{aligned} \tag{4.40}$$

where $u \in \mathbb{R}$ is the input and $\gamma[n]$ is the lower and upper temperature change bound ranging from 0.5 °C on day 0 to 2 °C on day 35. This does not restrict the permitted input space Ω_{k+1} for $k \to \infty$ as it changes with $u_{\kappa_k}[n]$.

Simulation Results

From Figure 4.12a it can be concluded that anchoring does not provide benefits under ideal modeling conditions, as I and I+A are almost identical – exactly as expected. However, anchoring is beneficial in conjunction with the data driven model, as D fails at minimizing FCR while D+A converges, but significantly slower than, e.g., I. This makes anchoring superior under data driven modeling conditions.

From I+MR on Figure 4.12b it can be concluded that using maximizing reference produces similar results to the unrealistic case where the smallest possible FCR is known. Also MR does not improve convergence conditions with a data driven model, since D+MR and D do not converge to zero error.

Using both MR and A, as shown in Figure 4.12c, leads to the conclusion that D+MR+A is the best performing implementable configuration of the algorithm, as D does not converge despite I and I+MR+A having superior performance. The convergence difference between I and D+MR+A is significant and is most notably caused by the measured weight bias $q_{w,\text{bias}}[n]$. To demonstrate that this is the case, removing the bias results in Figure 4.12d by enforcing $q_{w,\text{bias}}[n] = 0$, which has a slightly slower convergence rate compared to I and also a final FCR offset of ≈ 0.01.

In Figure 4.12e the D+MR+A algorithm is shown with different input change cost $W_{\Delta U}$, which demonstrates that if $W_{\Delta U}$ is configured too low, then the algorithm does not converge. Moreover, it suggests that iterative learning is required to solve the data driven FCR minimization problem and that TILC provides one possible solution.

4.2.5 Experimental Study

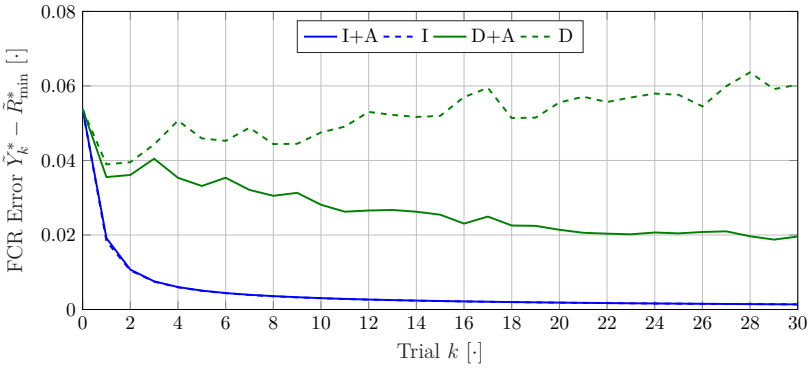
The results in this section are from an experimental study undertaken on the same location as Section 4.1 (FCR Optimization using Iterative Learning Control). Each batch approximately contains $40,000$ ROSS 308 broilers and an average duration of 34 days. A single trial conducted between June 27 and August 30, 2018, is detailed in the following.

Normalized FCR cost function

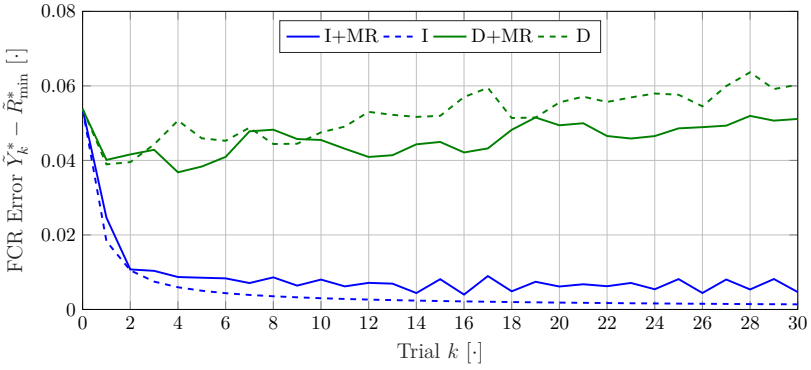
Batches have different durations, which makes FCR comparison difficult. For this reason, the FCR is normalized to the same weight ψ using the performance measure

$$J_{\text{FCR},\psi}(y_f, y_w) = \frac{y_f\left(1 - \frac{k_w}{\psi}\right) + y_w\left(\frac{k_f}{\psi}\right) - k_f}{y_w - k_w} \tag{4.41}$$

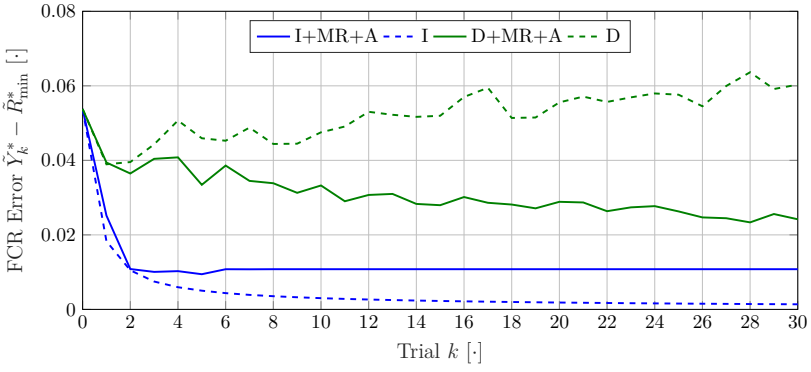
denoted FCR@2.2kg for convenience, where $y_f \in \mathbb{R}_+$ is the average feed consumed per broiler, $y_w \in \mathbb{R}_+$ is the average slaughter weight, $\psi = 2.2$ kg, and $k_w = -1.110$ kg and $k_f = -3.081$ kg are correction factors. This expression



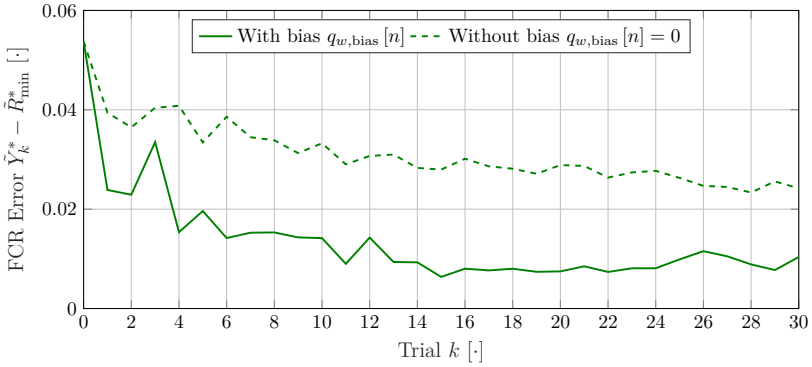
(a) Anchoring.



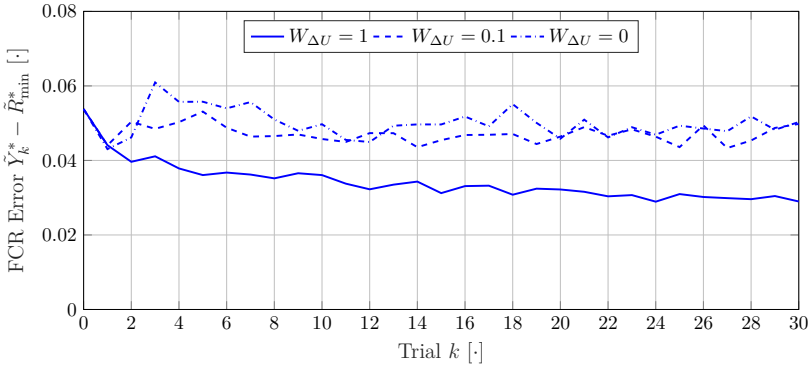
(b) Maximizing Reference.



(c) Anchoring and Maximizing Reference.



(d) D+MR+A with and without weight measurement bias $q_{w,\text{bias}}[n] = 0$.



(e) D+MR+A with different input change cost $W_{\Delta U}$ configurations.

Figure 4.12: Simulation results – see Section 4.2.4 for detailed explanations.

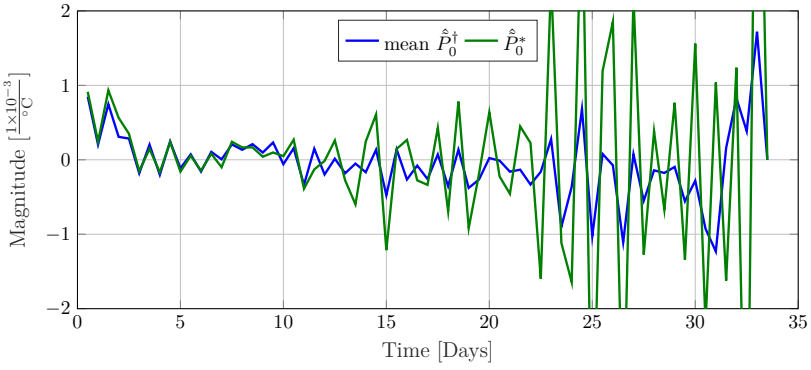


Figure 4.13: $\hat{\hat{P}}_{\kappa_k}$ and the mean $\hat{\hat{P}}_{\kappa_k}^\dagger$ for $k = 0$ of the experimental test.

is created using official regression formulas used by the Danish broiler industry[Det Danske Fjerkæraad, 2013, pp. 85], and replaces the augmented data driven model in (4.30) by

$$\tilde{Y}_k^*(U) = J_{\mathrm{FCR},\psi}(\tilde{Y}_{k,f}(U), \tilde{Y}_{k,w}(U)). \tag{4.42}$$

Note that (4.42) differs from the cost function in Section 4.1 (FCR Optimization using Iterative Learning Control).

"Extended" TILC

In Figure 4.13 the terminal system matrix $\hat{\tilde{P}}^*_{\kappa_k}$ for $k = 0$ is shown, which has a significant degree of "ripple" from day 21 and onwards. This feature is caused by ripples in the training data and falsely suggests that FCR can be decreased through temperature fluctuations. A straightforward solution, available within point-to-point ILC framework, is to *extend* the terminal ILC design to include the last $N_\dagger \in \mathbb{Z}_+$ output samples, i.e.,

$$Y_k^\dagger = \begin{bmatrix} y_k^*[N_e - N_\dagger + 1] & \cdots & y_k^*[N_e] \end{bmatrix}^T \in \mathbb{R}_+^{N_\dagger}. \tag{4.43}$$

The extended ILC problem now is

$$U_{k+1} = \underset{U \in \Omega_{k+1}}{\arg\min} \left\| \tilde{E}^\dagger_{\kappa_k}(U) \right\|^2_{W^\dagger_{\tilde{E}}} + \|U - U_{\kappa_k}\|^2_{W_{\Delta U}} \tag{4.44a}$$

subject to (4.36),

$$\tilde{R}_k^\dagger = \tilde{Y}_k^\dagger(U_k) - \mathcal{R}^\dagger, \tag{4.44b}$$

$$\tilde{E}^\dagger_{\kappa_k}(U) = \tilde{R}^\dagger_{\kappa_k} - \tilde{Y}^\dagger_{\kappa_k}(U) \text{ and} \tag{4.44c}$$

$$\tilde{Y}^\dagger_{\kappa_k}(U) = \hat{\tilde{P}}^\dagger_{\kappa_k} U + \hat{\tilde{K}}^\dagger_{\kappa_k} \tag{4.44d}$$

where $W^\dagger_{\tilde{E}} \in \mathbb{R}^{N_\dagger \times N_\dagger}$, $\tilde{R}^\dagger_k \in \mathbb{R}^{N_\dagger}$, $\hat{\tilde{P}}^\dagger_{\kappa_k} \in \mathbb{R}^{N_\dagger \times N_\dagger}$.

Method Configuration

The input variable selection algorithm selects up to 2 variables from the available potential inputs, e.g. CO2 denoted by $u_i[k \mid t]$ with index i, and up to 2 lags are selected per selected input, e.g. $u_i[k-1 \mid t]$ and $u_i[k-3 \mid t]$. The weight shape cost function is configured with N_ϕ = day 15, and the extended TILC is configured with $N_\dagger = 4$ samples. A total of $N_m = 64$ ensemble models are used, of which the remaining settings are identical to the simulation study as described in Section 4.2.4.

Experimental Results

Figure 4.15 shows relevant measured signals for $k = 1$, where the FCR@2.2kg of trial $k = 1$ is approximately 6% smaller compared to $k = 0$. The terminal broiler weight is 200g higher and the terminal cumulative feed consumption is only 100g higher, which is a disproportionate exchange rate. The initial input change is approximately 0.5 °C lower for days 0–4 and 9–15, and approximately 2 °C higher for day 27. The initial decrease in temperature decreased the broiler growth rate, as the operator reported mild signs of cold stress in the broilers on visual inspection.

Applying the new algorithm results in a FCR@2.2kg decrease of 5.9% (0.059) and FCR decrease of 1.4% (0.014) for trial $k = 1$ – calculated using the slaughter weight. In Figure 4.14 the historic performance of the house is given, which shows that trial $k = 1$ has a very promising historically low FCR. This result is very close to the trial-to-trial FCR decrease for the first trial of the simulation study in Figure 4.12 with a FCR decrease of approximately 1% (0.01).

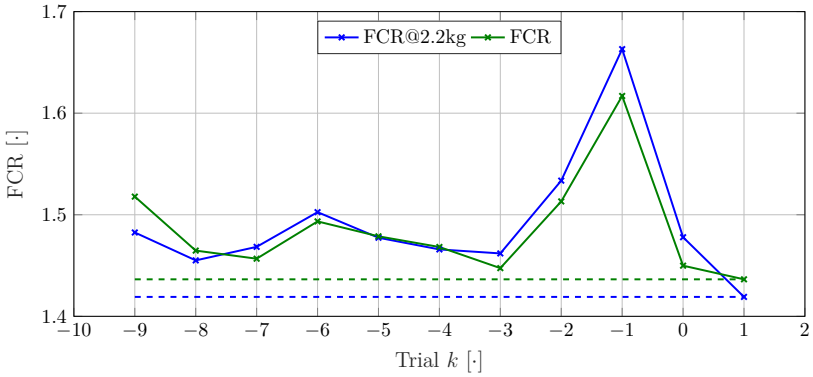


Figure 4.14: FCR and FCR@2.2kg performance overview of the recent 10 trials $k \in \{-9, \ldots, 0\}$ and the current trial $k = 1$. Trial $k \in \{-1, -2\}$ have unusually high FCR due to an unusually cold winter, rendering the temperature regulation unable to maintain the desired temperature. Trial $k \in \{-4, -3, -2, -1\}$ are the 4 trials detailed in Section 4.1.

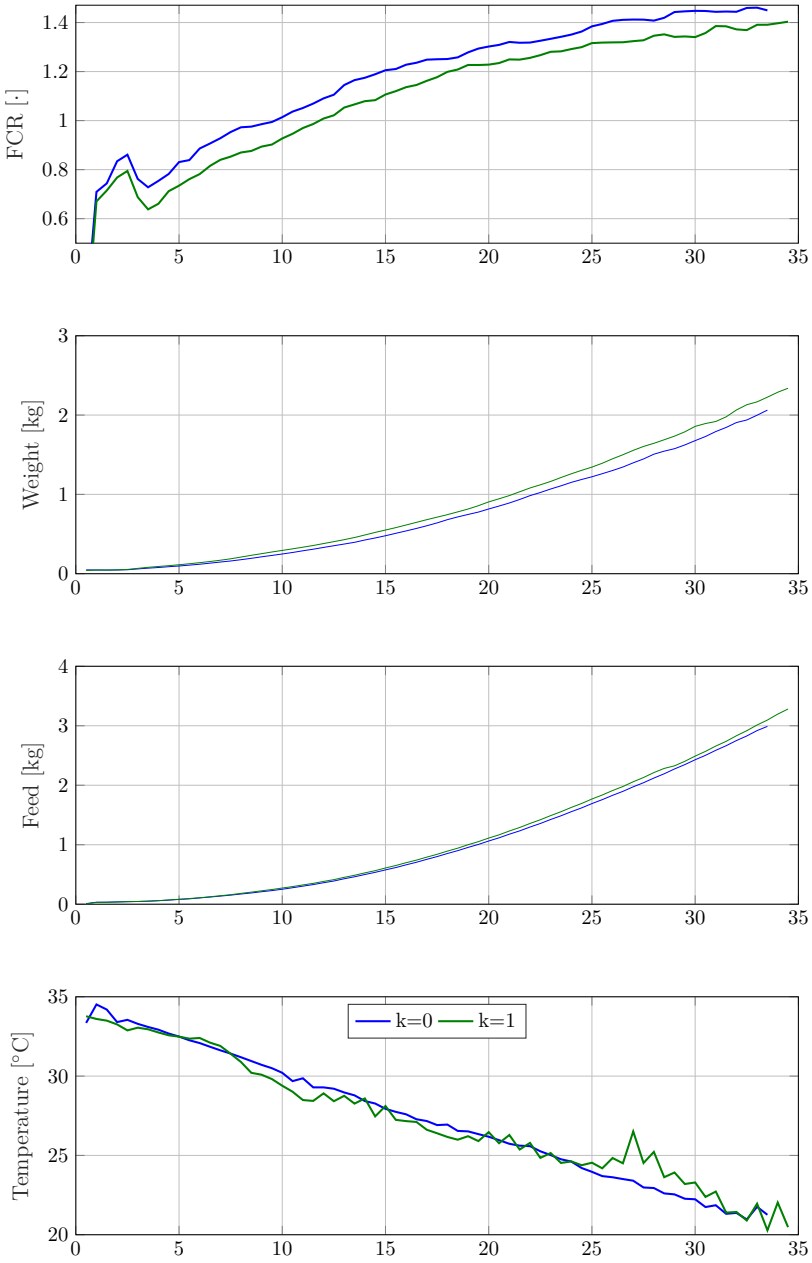


Figure 4.15: Experimental results for $k = 1$ using the new algorithm. The FCR, broiler weight, feed consumption and measured temperature is depicted for trial $k \in \{0, 1\}$ along with their difference in red.

4.3 Economical Considerations

This section reflects on the potential economic impact of applying the two developed FCR minimization algorithms. The potential savings summary is presented in Table 4.2, and gives a naive view of the savings potential of the algorithm. Note that the average global FCR is much higher than the nominal FCR of 1.470 in this study, e.g., the average FCR in USA is 1.83[National Chicken Council, 2018b]. The absolute numbers cannot be considered very accurate, as sentient biological production can sometimes give misleading results. But it serves as a clear indication for broiler FCR minimization being a worthwhile economic endeavor. It also shows that the performance index, i.e. FCR and FCR@2.2kg, have a huge impact on the potential savings, suggesting that one should be cautious when comparing FCR between farms. Nonetheless, these experimental results demonstrate the basic feasibility of the algorithms and provide a basis for onward development.

	FCR@2.2kg	FCR	
Slaughter Weight	2.2	2.19	[kg]
Nominal FCR	1.467	1.470	[kg/kg]
Potential FCR	1.407	1.432	[kg/kg]
Potential FCR Difference	6.0	3.8	[%]
House Feed Savings	48,200	26,700	[kg/year]
House Feed Cost Savings	13,512	8,571	[€/year]
Global Feed Savings	3.84	6.03	[billion kg/year]
Global Feed Cost Savings	1.93	1.23	[billion €/year]

Table 4.2: Potential impact summary by applying the two FCR optimization algorithms in terms of FCR@2.2kg and FCR presented on Figure 4.14. In Denmark, feed cost of 0.32€/kg is fairly typical. The test broiler house executes 8.3 batches per year with 38,000 broilers and slaughter weight of 2.19kg on average. The global broiler consumption is assumed to be 83% of 120×10^9kg, as discussed in Chapter 1 (Introduction).

5 Concluding Remarks

This chapter concludes on the research objectives stated in Section 1.3 and provides future perspectives and reflections.

The primary objective of this study is to both minimize the environmental impact and maximize the profit margin of broiler production though suitably designed climate control. Specifically, to implement an "automatic farmer assistant" that gradually learns the batch production process – enabling it to improve production performance batch-to-batch.

The first research objective was to develop a data driven dynamic broiler growth model, which was addressed in Chapter 3 (Data Driven Broiler Modeling) and by the contributions of Paper A, Paper B and Paper C. In this study, dynamic neural network (DNN) models are successfully trained on industrial scale production data. Ensemble DNN models are successfully used to forecast broiler weight on ≈4 years of production data using environmental broiler house variables, where input data from a prior batch is used as a substitute for unknown missing future input values. An input variable selection algorithm is used to automatically determine the DNN input-output structure, and investigate the previously unexplored temporal relationship between environmental broiler house variables and broiler growth, feed and water consumption. Evidence for the environmental broiler house variables having a significant dynamic influence on broiler growth, feed and drinking behavior was established. To better cope with the broiler weight measurement bias, the accurately measured slaughter weight is successfully included in the model, of which the models slaughter weight forecasting ability is greatly enhanced. Insights into effective algorithm configuration are also investigated, which among others suggest that at least 6 training batches are required.

The second research objective was to optimize broiler production using Iterative Learning Control (ILC), which was addressed in Chapter 4 (Broiler Feed Conversion Rate Optimization) and by the contributions of Paper D, Paper E and Paper F. In this study, this has exclusively been done through feed conversion rate (FCR) minimization by regulating the broiler house temperature. An ILC algorithm was used in conjunction with the DNN model without broiler measurement bias compensation, resulting in a FCR reduction of 2.4% over 2 successful trials. A modified Terminal ILC algorithm was used in conjunction with the DNN model with broiler measurement bias compensation, resulting in a FCR reduction of 1.4% over a single trial. Traditional Terminal ILC (TILC) is modified by turning it into a descent type algorithm, i.e. unsuccessful optimization attempts are discarded, to better cope with the uncertain nature of the data driven DNN model, and a lack of a known minimum FCR reference is remedied by improving performance trial-to-trial. A heuristic broiler growth model was formulated to assist broiler optimization development in simulation,

which most notably was used to demonstrate that the modifications are necessary for FCR minimization and that an iterative searching strategy, such as ILC, is required for FCR minimization. The heuristic broiler growth model was also used to demonstrate that the broiler measurement bias significantly influence performance, and that the DNN model with broiler measurement bias compensation copes with this in simulation, which might also be the case in practice. A combined total FCR reduction of 3.8% was obtained for the two FCR optimization attempts – corresponding to a yearly saving of 8.500€ and 26.7 tones of broiler feed per broiler house.

Future Perspectives and Reflections

It is the authors personal opinion that neural networks and the like have been drastically overestimated and romanticized as a universal solution to any intractable or hard problem in recent years. Application specific knowledge must somehow be embedded into algorithms to facilitate a reasonable chance of success – like the slaughter weight inclusion to compensate for the measurement bias. Many unaddressed problems still exist, especially given the fact that this study is the first to investigate FCR optimization by regulating environmental climate house conditions. For convenience the future perspectives are ordered from short to long term, thus presenting a plausible roadmap going forward.

Most notably, the developed algorithms appear to be successful in the short term on a single location, of which its long term performance on many locations still remains to be investigated. It is particularly interesting to investigate the performance influence of the broiler house size, ventilation means, house deterioration, and outside climate as such factors greatly influence the degree to which the broiler house climate can be maintained. These factors are characterized by a "slow" or bounded rate of change trial-to-trial, which ILC is expected to suppress satisfactory, although slightly worse performance is inevitable.

Random elements that alter the initial conditions trial-to-trial are known to cause problems for ILC performance. Parent stock age; that governs growth speed and initial broiler weight, changing batch start times; that changes broiler age on a certain day, nondeterministic broiler management; that results in unmeasured deviations, and sickness; which temporarily stunts growth, give rise to the largest concern for the developed TILC algorithm. These disturbances can directly be dealt with through feedback, where the control signal $u[n]$ at sample n is a function of broiler weight $m[n]$, i.e. $u[n] = f(m[n])$. Thus, making it a gain scheduling ILC with broiler weight as scheduling parameter – this has to the best of the authors knowledge not been subject to much research in scientific literature, perhaps because it lacks a practical application. In this way, the temperature remains "optimal" for a class of random unmodeled disturbances regardless of broiler maturity deviations from the prior trial. However, this requires the broiler weight measurement bias problem to be solved, or perhaps the inclusion of time consuming manual control measurements. This could be facilitated by something as simple as a lookup table

of the ILC control signal, which is unique for a given weight as broiler weight is monotonically increasing and the ILC control signal can be designed to be either monotonically decreasing or increasing.

The current cross validation scheme can most notably be described as a "leave-one-batch-out" cross validation method with a fixed "left-out" batch. An ensemble model based on an exhaustive combination of the "left-out" batch is expected to result in a significantly less biased model. Network weight pruning techniques, such as the optimal brain surgeon algorithm, are subsequently expected to improve performance by reducing unnecessary model complexity. Sigmoidal-like activation functions, like atan used in this study, are expected to produce the best results for broiler growth modeling, as they are often used for growth curve modeling. However, alternative neural network structures could still be worth looking into to better favor other aspects of broiler production; like the influence of temperature and light, where radially bounded networks could be a good starting point.

Different input variable selection (IVS) approaches could be investigated to improve the DNN model structure correctness. Investigation of different kernel density estimation (KDE) techniques used for regression based partial mutual information (PMI) is perhaps the best candidate for improvement, as it could decrease the number of model parameters. Simple but proven kernel bandwidth rules where used as a basis for KDE in this study, but adaptive or more sophisticated bandwidth estimation techniques could perhaps increase accuracy – like the copula entropy KDE. *Support output variable selection* (SOVS) could most notably prove useful in data driven dynamic modeling applications. The mutual information (MI) explained of the input Y by the input candidate S through the support output candidate Z that is not explained by the previously selected inputs U can be calculated by $I(S;Y \mid U;Z)$. This requires high order KDE, which can be estimated using lower order estimation techniques, e.g., by relaxing some MI assumptions[Vinh et al., 2016] in combination with regression based PMI. In this way, a "compact" graph of support outputs and inputs that best explain the time dependence of the primary output is created. SOVS has to the best of the authors knowledge not been investigated in scientific literature. However, as with MI-based IVS, a proper stopping criteria is expected to be difficult to determine.

Other potential performance indexes and input variables can be investigated, of which examples are provided in the order that the author think is most likely to succeed. 1) Optimizing FCR using both temperature and light intensity, where light intensity is known to be highly influential on broiler growth. 2) Include weather forecasts to reduce future uncertainty. 3) The broiler body temperature could automatically by measured using wireless tags implanted in a significant number of the broiler population and be used to maintain an optimal body temperature through climate regulation. 4) Maximize overall batch profit by including broiler and feed cost in the cost function. 5) Animal welfare can be optimized by including "health" related indicators in the cost function, such as litter ammonia development that is known to cause hook burns, i.e.

acid burns on the feet, and is expected to be indirectly measured by the air ammonia concentration.

A sufficient quantity of high quality data is a prerequisite for high precision data driven modeling, which ideally should cover all relevant input and output frequencies and amplitudes. This is *very hard* to facilitate in practice, especially considering that broiler production deals with live animals where some input combinations are strictly prohibited. Adding noise to the input signal could drastically increase data quality, but most likely decrease short term performance, which presents a dilemma. Currently, only a somewhat limited number of broiler farmers value data accumulation, which subsequently limits the number of broiler houses where the algorithm can be applied immediately. Furthermore, at least 6 batches was found to be required in this study – corresponding to ≈9 months of data. For this reason, a "similarity"-index could be used to automatically identify a portfolio of similar broiler houses, which could be used to "prime" the algorithm with data when only a few batches are available. This is expected to increase early data quantity and potentially increase the initial convergence rate. The portfolio could also be used for outlier-batch data identification, which most notably increase data quality.

In this context, a large scale portfolio model could perhaps yield faster convergence rate of the TILC algorithm, and open the possibility of detecting rare events using fault tolerant control techniques, e.g., impending sickness, failing sensors and management mistakes. In this situation, distributed privacy preserving algorithms could prove to be a valuable tool to increase collaboration between farmers with minimal individual risk, i.e., a global solution based on private data from many agents is found without revealing private data to other agents. Given the fairly standard algorithms used in study, this approach appears as a viable solution. This could ultimately inspire a completely decentralized broiler management system, without the need for competing large scale integrators, that automatically optimize the most important aspects of the broiler supply chain, as mentioned in Section 1.1, to meet the market demand in terms of price, meat quality, and animal welfare.

Lastly, the developed techniques are not limited to broiler production, but can be applied to any food production that has a mature data acquisition tradition with some application dependent modifications. Broiler production has a fairly mature data acquisition tradition, but key variables such as automatic broiler weight and feed consumption measurements required for production optimization still have low priority for a majority of broiler farmers, as they only provide indirect value. If having these key variables could guarantee safe and automatic production optimization, it would provide a clear incentive to adopt a mature data acquisition tradition. This presents a "chicken or the egg causality dilemma", where broiler data acquisition tradition, production optimization options and broiler farmers co-develop, which greatly influence the long term success of automatic broiler optimization.

References

L. H. P. Abreu, T. Y. Junior, É. J. Fassani, A. T. Campos, and D. Lourençoni. "Fuzzy modeling of broiler performance, raised from 1 to 21 days, subject to heat stress". *Engenharia Agrícola*, 35(6):967–978, dec 2015. doi:10.1590/1809-4430-eng.agric.v35n6p967-978/2015.

J. M. Aerts, M. Lippens, G. De Groote, J. Buyse, E. Decuypere, E. Vranken, and D. Berckmans. "Recursive prediction of broiler growth response to feed intake by using a time-variant parameter estimation method". *Poultry Science*, 82:40–49, 2003. ISSN 0032-5791.

S. Aggrey. "Comparison of three nonlinear and spline regression models for describing chicken growth curves". *Poultry Science*, 81(12):1782–1788, dec 2002. doi:10.1093/ps/81.12.1782.

M. Aghbashlo, S. Hosseinpour, and A. S. Mujumdar. "Application of artificial neural networks (ANNs) in drying technology: A comprehensive review". *Drying Technology*, 33(12):1397–1462, may 2015. doi:10.1080/07373937.2015.1036288.

L. A. Aguirre and C. Letellier. "Modeling nonlinear dynamics and chaos: A review". *Mathematical Problems in Engineering*, 2009:to, 2018a. doi:10.1155/2009/238960.

L. A. Aguirre and C. Letellier. "Modeling nonlinear dynamics and chaos: A review". *Mathematical Problems in Engineering*, 2009:to, 2018b. doi:10.1155/2009/238960.

H. Ahmadi and M. Mottaghitalab. "Hyperbolastic models as a new powerful tool to describe broiler growth kinetics". *Poultry Science*, 86(11):2461–2465, nov 2007. doi:10.3382/ps.2007-00086.

H.-S. Ahn, Y. Chen, and K. Moore. "Iterative learning control: Brief survey and categorization". *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1099 –1121, 2007.

N. Amann, D. H. Owens, and E. Rogers. "Iterative learning control using optimal feedback and feedforward actions". *International Journal of Control*, 65(2):277–293, sep 1996. doi:10.1080/00207179608921697.

S. Amraei, S. A. Mehdizadeh, and S. Salari. "Broiler weight estimation based on machine vision and artificial neural network". *British Poultry Science*, 58(2):200–205, mar 2017. doi:10.1080/00071668.2016.1259530.

S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan. "Medical image analysis using convolutional neural networks: A review". *Journal of Medical Systems*, 42(11):226, Oct 2018. ISSN 1573-689X. doi:10.1007/s10916-018-1088-1.

S. Arimoto, S. Kawamura, and F. Miyazaki. "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems". In *The 23rd IEEE Conference on Decision and Control*. IEEE, dec 1984. doi:10.1109/cdc.1984.272176.

S. Arlot and A. Celisse. "A survey of cross-validation procedures for model selection". *Statistics Surveys*, 4(0):40–79, 2010. doi:10.1214/09-ss054.

Aviagen. *ROSS Environmental Management in the Broiler House*, 2010. URL: http://en.aviagen.com/tech-center/download/236/Ross_Environmental_Management_in_the_Broiler_House.pdf.

Aviagen. *Ross 308 Broiler: Performance Objectives*, 2014a. URL: http://en.aviagen.com/assets/Tech_Center/Ross_Broiler/Ross-308-Broiler-PO-2014-EN.pdf.

Aviagen. *ROSS Broiler Management Handbook*, 2014b. URL: http://en.aviagen.com/assets/Tech_Center/Ross_Broiler/Ross-Broiler-Handbook-2014i-EN.pdf.

P. Barmpalexis, A. Karagianni, G. Karasavvaides, and K. Kachrimanis. "Comparison of multi-linear regression, particle swarm optimization artificial neural networks and genetic programming in the development of mini-tablets". *International Journal of Pharmaceutics*, 551:166 to 176, 11 2018. doi:10.1016/j.ijpharm.2018.09.026.

J. Beirlant, E. J. Dudewicz, L. Györfi, and E. C. Van Der Meulen. "Nonparametric entropy estimation: An overview". *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.5281&rep=rep1&type=pdf.

W. Bessei. "Welfare of broilers: a review". *World's Poultry Science Journal*, 62(03): 455–466, sep 2006. doi:10.1079/wps2005108.

J. Blahová, R. Dobšíková, E. Straková, and P. Suchý. "Effect of low environmental temperature on performance and blood system in broiler chickens (gallus domesticus)". *Acta Veterinaria Brno*, 76(8):S17–S23, 2007. doi:10.2754/avb200776s8s017.

J. Bolder and T. Oomen. "Data-driven optimal ILC for multivariable systems: Removing the need for l and q filter design". In *2015 American Control Conference (ACC)*. IEEE, jul 2015. doi:10.1109/acc.2015.7171880.

D. Bonné, B. Sten, and S. Jørgensen. "Data-driven modeling of batch processes". *IFAC Proceedings Volumes*, 37(1):589–594, 01 2004.

G. J. Bowden, H. R. Maier, and G. C. Dandy. "Input determination for neural network models in water resources applications. part 2. case study: forecasting salinity in a river". *Journal of Hydrology*, 301(1-4):93–107, jan 2005. doi:10.1016/j.jhydrol.2004.06.020.

D. A. Bristow, M. Tharayil, and A. G. Alleyne. "A survey of iterative learning control". *Control Systems, IEEE*, 26(3):96 – 114, 2006.

G. Brown, A. Pocock, M. jie Zhao, M. Luján, and I. Guyon. "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection". *Journal of Machine Learning Research*, 13, 2012.

F. Burden and D. Winkler. "Bayesian regularization of neural networks". In *Methods in Molecular Biology*, pages 23–42. Springer Science + Business Media, 2008. doi:10.1007/978-1-60327-101-1_3.

M. G. L. Cândido, I. D. F. F. Tinôco, F. de A. de C. Pinto, N. T. Santos, and R. P. Roberti. "Determination of thermal comfort zone for early-stage broilers". *Engenharia Agrícola*, 36(5):760–767, oct 2016. doi:10.1590/1809-4430-eng.agric.v36n5p760-767/2016.

O. Cangar, J.-M. Aerts, E. Vranken, and D. Berckmans. "End-weight prediction in broiler growth". *British Poultry Science*, 47(3):330–335, jun 2006. doi:10.1080/00071660600741735.

O. Cangar, J.-M. Aerts, E. Vranken, and D. Berckmans. "Online growth control as an advance in broiler farm management". *Poultry Science*, 86:439–443, 2007. ISSN 0032-5791.

D. C. Cassuce, I. de F. F. Tinôco, F. C. Baêta, S. Zolnier, P. R. Cecon, and M. de F. A. Vieira. "Thermal comfort temperature update for broiler chickens up to 21 days of age". *Engenharia Agrícola*, 33(1):28–36, feb 2013. doi:10.1590/s0100-69162013000100004.

A. Chedad, E. Vranken, J.-M. Aerts, and D. Berckmans. "Behaviour of chickens towards automatic weighing systems". *IFAC Proceedings Volumes*, 33(29):207 – 212, 2000. doi:https://doi.org/10.1016/S1474-6670(17)36778-2.

A. Chedad, J.-M. Aerts, E. Vranken, M. Lippens, J. Zoons, and D. Berckmans. "Do heavy broiler chickens visit automatic weighing systems less than lighter birds?". *British Poultry Science*, 44(5):663–668, dec 2003. doi:10.1080/00071660310001643633.

L. Chen, L. Ye, V. Singh, J. Zhou, and S. Guo. "Determination of input for artificial neural networks for flood forecasting using the copula entropy method". *Journal of Hydrologic Engineering*, 19(11):04014021, nov 2014. doi:10.1061/(asce)he.1943-5584.0000932.

A. Chlingaryan, S. Sukkarieh, and B. Whelan. "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review". *Computers and Electronics in Agriculture*, 151:to, 08 2018. doi:10.1016/j.compag.2018.05.012.

B. Chu, C. T. Freeman, and D. H. Owens. "A novel design framework for point-to-point ILC using successive projection". *IEEE Transactions on Control Systems Technology*, 23(3):1156–1163, may 2015. doi:10.1109/tcst.2014.2356931.

Cobb-Vantress. *Cobb 500 broiler performance & nutrition supplement*, 2018. URL: `http://www.cobb-vantress.com/docs/default-source/cobb-500-guides/Cobb500_Broiler_Performance_And_Nutrition_Supplement.pdf`.

O. E. de Noord. "The influence of data preprocessing on the robustness and parsimony of multivariate calibration models". *Chemometrics and Intelligent Laboratory Systems*, 23(1):65–70, apr 1994. doi:10.1016/0169-7439(93)e0065-c.

M. L. de V. Queiroz, J. A. D. B. Filho, L. M. Duarte, D. de F. Brasil, and C. R. F. Gadelha. "Environmental and physiological variables during the catching of broilers". *Revista Brasileira de Ciência Avícola*, 17(1):37–44, mar 2015. doi:10.1590/1516-635x170137-44.

E. Decuypere, K. Tona, V. Bruggeman, and F. Bamelis. "The day-old chick: a crucial hinge between breeders and broilers". *World's Poultry Science Journal*, 57 (02):127–138, jun 2001. doi:10.1079/wps20010010.

T. Demmers, Y. Cao, S. Gauss, J. Lowe, D. Parsons, and C. Wathes. "Neural predictive control of broiler chicken growth". *IFAC Proceedings Volumes*, 43(6):311–316, 2010. doi:10.3182/20100707-3-be-2012.0061.

L. F. Demuner, D. Suckeveris, J. A. Muñoz, V. C. Caetano, C. G. de Lima, D. E. de Faria Filho, and D. E. de Faria. "Adjustment of growth models in broiler chickens". *Pesquisa Agropecuária Brasileira*, 52(12):1241–1252, dec 2017. doi:10.1590/s0100-204x2017001200013.

Det Danske Fjerkæraad. *Årsberetning 2013*. 2013. https://danskfjerkrae.dk/om-fjerkraebranchen/det-danske-fjaerkraeraad/aarsberetning/aarsberetning-2013.

L. Devroye and T. Wagner. "Distribution-free performance bounds for potential function rules". *IEEE Transactions on Information Theory*, 25(5):601–604, sep 1979. doi:10.1109/tit.1979.1056087.

A. Diez-Olivan, X. Averós, R. Sanz, B. Sierra, and I. Estevez. "Quantile regression forests-based modeling and environmental indicators for decision support in broiler farming". *Computers and Electronics in Agriculture*, mar 2018. doi:10.1016/j.compag.2018.03.025. in press.

J. Ding, B. Cichy, K. Galkowski, E. Rogers, and H.-Z. Yang. "Robust fault-tolerant iterative learning control for discrete systems via linear repetitive processes theory". *International Journal of Automation and Computing*, 12(3):254–265, jun 2015. doi:10.1007/s11633-015-0883-0.

V. M. dos Santos, B. S. L. Dallago, A. M. C. Racanicci, Â. P. Santana, and F. E. M. Bernal. "Effects of season and distance during transport on broiler chicken meat". *Poultry Science*, 96(12):4270–4279, oct 2017. doi:10.3382/ps/pex282.

E. N. Drăgoi, S. Curteanu, and D. Fissore. "On the use of artificial neural networks to monitor a pharmaceutical freeze-drying process". *Drying Technology*, 31(1):72–81, jan 2013. doi:10.1080/07373937.2012.718308.

K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer Science + Business Media, 2014. doi:10.1007/978-1-4471-5571-3.

C. A. Duran-Villalobos and B. Lennox. "Iterative learning modelling and control of batch fermentation processes". *IFAC Proceedings Volumes*, 46(32):511–516, dec 2013. doi:10.3182/20131218-3-in-2045.00010.

A. Famili, W.-M. Shen, R. Weber, and E. Simoudis. "Data preprocessing and intelligent data analysis". *Intelligent Data Analysis*, 1(1-4):3–23, 1997. doi:10.1016/s1088-467x(98)00007-9.

M. R. Farag and M. Alagawany. "Physiological alterations of poultry to the high environmental temperature". *Journal of Thermal Biology*, 76:101–106, aug 2018. doi:10.1016/j.jtherbio.2018.07.012.

M. Fatehnia and G. Amirinia. "A review of genetic programming and artificial neural network applications in pile foundations". *International Journal of Geo-Engineering*, 9, 01 2018. doi:10.1186/s40703-017-0067-6.

P. R. Ferket and A. G. Gernat. "Factors that affect feed intake of meat birds: A review". *International Journal of Poultry Science*, 5(10):905–911, oct 2006. ISSN 1682-8356. doi:10.3923/ijps.2006.905.911.

P. F. P. Ferraz, T. Y. Junior, R. R. de Lima, G. A. e Silva Ferraz, and H. Xin. "Performance of chicks subjected to thermal challenge". *Pesquisa Agropecuária Brasileira*, 52(2):113–120, feb 2017. doi:10.1590/s0100-204x2017000200005.

P. F. P. Ferraz, T. Y. Junior, Y. F. Hernandez-Julio, G. A. e S. Ferraz, M. A. J. G. Silva, and F. A. Damasceno. "Genetic fuzzy system for prediction of respiratory rate of chicks subject to thermal challenges". *Revista Brasileira de Engenharia Agrícola e Ambiental*, 22(6):412–417, jun 2018. doi:10.1590/1807-1929/agriambi.v22n6p412-417.

I. Fontana, E. Tullo, L. Carpentier, D. Berckmans, A. Butterworth, E. Vranken, T. Norton, D. Berckmans, and M. Guarino. "Sound analysis to model weight of broiler chickens". *Poultry Science*, 96(11):3938–3943, sep 2017. doi:10.3382/ps/pex215.

C. T. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes, and K. L. Meadmore. *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation*. Springer, 2015. ISBN 978-1-4471-6725-9. doi:10.1007/978-1-4471-6726-6.

K. Furuta and M. Yamakita. "A design of learning control system for linear multivariable systems". *Transactions of the Society of Instrument and Control Engineers*, 22(12):1248–1255, 1986. doi:10.9746/sicetr1965.22.1248.

K. Gibert, M. Sànchez–Marrè, and J. Izquierdo. "A survey on pre-processing techniques: Relevant issues in the context of environmental data mining". *AI Communications*, 29(6):627–663, Dec 2016. ISSN 1875-8452. doi:10.3233/AIC-160710.

M. Günal. "The effects of early-age thermal manipulation and daily short-term fasting on performance and body temperatures in broiler exposed to heat stress". *Journal of Animal Physiology and Animal Nutrition*, pages no–no, jul 2012. doi:10.1111/j.1439-0396.2012.01330.x.

J. Grahovac, A. Jokić, J. Dodić, D. Vučurović, and S. Dodić. "Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks". *Renewable Energy*, 85:953–958, jan 2016. doi:10.1016/j.renene.2015.07.054.

M. Hagan and M. Menhaj. "Training feedforward networks with the marquardt algorithm". *IEEE Transactions on Neural Networks*, 5:989 to 993, 1994. doi:10.1109/72.329697.

J. A. Hamidu, C. A. Torres, M. L. Johnson-Dahl, and D. R. Korver. "Physiological response of broiler embryos to different incubator temperature profiles and maternal flock age during incubation. 1. embryonic metabolism and day-old chick quality". *Poultry Science*, 97(8):2934–2946, jul 2018. doi:10.3382/ps/pey089.

A. e. F. N. o. Hasan Eleroğlu, Arda Yıldırım and M. Duman. “Comparison of growth curves by growth models in slow–growing chicken genotypes raised the organic system”. *International Journal of Agriculture and Biology*, 16(3):529–535, 2014.

B. Hassibi and D. G. Stork. “Second order derivatives for network pruning: Optimal brain surgeon”. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 164–171. Morgan-Kaufmann, 1993. URL: `http://papers.nips.cc/paper/647-second-order-derivatives-for-network-pruning-optimal-brain-surgeon.pdf`.

B. Hassibi, D. G. Stork, and G. J. Wolff. “Optimal brain surgeon and general network pruning”. In *IEEE International Conference on Neural Networks*. IEEE, 1993. doi:10.1109/icnn.1993.298572.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.

J. He, C. Valeo, A. Chu, and N. F. Neumann. “Prediction of event-based stormwater runoff quantity and quality by ANNs developed using PMI-based input selection”. *Journal of Hydrology*, 400(1-2):10–23, mar 2011. doi:10.1016/j.jhydrol.2011.01.024.

S. Henriksen, T. Bilde, and A. B. Riber. “Effects of post-hatch brooding temperature on broiler behavior, welfare, and growth”. *Poultry Science*, 95(10):2235–2243, jul 2016. doi:10.3382/ps/pew224.

P. A. Henríquez and G. A. Ruz. “A non-iterative method for pruning hidden neurons in neural networks with random weights”. *Applied Soft Computing*, 70:1109–1121, sep 2018. doi:10.1016/j.asoc.2018.03.013.

S. C. Hernandez, J. A. Bueno, E. N. Sanchez, and L. Diaz-Jimenez. “State estimation by artificial neural networks in a continuous bioreactor”. *IFAC Proceedings Volumes*, 46(31):215–220, 2013. doi:10.3182/20131216-3-in-2044.00033.

F. Hippenstiel, A. Abdel-Wareth, S. Kehraus, and K.-H. Südekum. “Effects of selected herbs and essential oils, and their active components on feed intake and performance of broilers – a review”. *Archiv fur Geflugelkunde*, 75:226–234, 10 2011.

B. O. Hughes. “The principles underlying choice feeding behaviour in fowls—with special reference to production experiments”. *World's Poultry Science Journal*, 40 (02):141–150, jun 1984. doi:10.1079/wps19840012.

P. Hunton. “100 years of poultry genetics”. *World's Poultry Science Journal*, 62(03): 417–428, sep 2006. doi:10.1079/wps2006104.

A. Ipek and A. Sozcu. “The effects of eggshell temperature fluctuations during incubation on welfare status and gait score of broilers”. *Poultry Science*, 95(6):1296–1303, mar 2016. doi:10.3382/ps/pew056.

A. Ipek, U. Sahan, S. C. Baycan, and A. Sozcu. “The effects of different eggshell temperatures on embryonic development, hatchability, chick quality, and first-week broiler performance”. *Poultry Science*, 93(2):464–472, jan 2014. doi:10.3382/ps.2013-03336.

J. J. E. Dennis and J. J. Moré. "Quasi-newton methods, motivation and theory". *SIAM Review*, 19(1):46–89, jan 1977. doi:10.1137/1019005.

M. Jabri and B. Flower. "Weight perturbation: an optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks". *IEEE Transactions on Neural Networks*, 3(1):154–157, 1992. doi:10.1109/72.105429.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

S. V. Johansen, M. R. Jensen, B. Chu, J. D. Bendtsen, and E. Rogers. "Broiler growth optimization using norm optimal terminal iterative learning control". *Proceedings of Control Technology and Applications*, 2018.

S. V. Johansen, J. D. Bendtsen, and J. Mogensen. "Broiler slaughter weight forecasting using dynamic neural network models". *Proceedings of the International Conference on Industrial Engineering and Applications*, 2019a. doi:10.1109/IEA.2019.8714850.

S. V. Johansen, J. D. Bendtsen, and J. Mogensen. "Broiler growth optimization using optimal iterative learning control". *Proceedings of the American Control Conference*, 2019b.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Broiler weight forecasting using dynamic neural network models with input variable selection". *Journal of Computers and Electronics in Agriculture*, 2019c. doi:10.1016/j.compag.2018.12.014.

S. V. Johansen, M. R. Jensen, B. Chu, J. D. Bendtsen, J. Mogensen, and E. Rogers. "Broiler FCR optimization using norm optimal terminal iterative learning control". *IEEE Transactions on Control Systems Technology*, 2019d. In review.

J. Kennedy and R. Eberhart. "Particle swarm optimization". *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995. doi:10.1109/ICNN.1995.488968.

K. Kuźniar and M. Zając. "Some methods of pre-processing input data for neural networks". *Computer Assisted Methods in Engineering and Science*, 22(2):141–151, 2015.

A. Laudani, G. M. Lozito, F. R. Fulginei, and A. Salvini. "On training efficiency and computational costs of a feed forward neural network: A review". *Computational Intelligence and Neuroscience*, 2015:1–13, 2015. doi:10.1155/2015/818243.

S. Lertworasirikul and Y. Tipsuwan. "Moisture content and water activity prediction of semi-finished cassava crackers from drying process with artificial neural network". *Journal of Food Engineering*, 84(1):65–74, jan 2008. doi:10.1016/j.jfoodeng.2007.04.019.

X. Li, H. R. Maier, and A. C. Zecchin. "Improved PMI-based input variable selection approach for artificial neural network and other data driven environmental and water resource models". *Environmental Modelling & Software*, 65:15–29, mar 2015. doi:10.1016/j.envsoft.2014.11.028.

K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis. "Machine learning in agriculture: A review". *Sensors*, 18:2674 to, 08 2018. doi:10.3390/s18082674.

X. Liang, A. M. Javid, M. Skoglund, and S. Chatterjee. "Distributed large neural network with centralized equivalence", 04 2018.

I. Lim and K. L. Barton. "Pareto iterative learning control: Optimized control for multiple performance objectives". *Control Engineering Practice*, 26:125–135, may 2014. doi:10.1016/j.conengprac.2014.01.011.

I. Lim, D. J. Hoelzle, and K. L. Barton. "A multi-objective iterative learning control approach for additive manufacturing applications". *Control Engineering Practice*, 64(11):74–87, 2017.

H. Lin, H. C. Jiao, J. Buyse, and E. Decuypere. "Strategies for preventing heat stress in poultry". *World's Poultry Science Journal*, 62(01):71–86, mar 2006. doi:10.1079/wps200585.

A. Z. Lopes, L. Ferreira, T. Y. Junior, and W. S. Lacerda. "Modeling productive performance of broiler chickens with artificial neural network". In *Livestock Environment VIII, 31 August - 4 September 2008, Iguassu Falls, Brazil*. American Society of Agricultural and Biological Engineers (ASABE), 2008. doi:10.13031/2013.25602.

B. Luong. "Free-knot spline approximation". MATLAB Central File Exchange. Retrieved 29 June 2017, `https://se.mathworks.com/matlabcentral/fileexchange/25872-free-knot-spline-approximation`.

C. M. Maatjens, I. A. M. van Roovert-Reijrink, B. Engel, C. W. van der Pol, B. Kemp, and H. van den Brand. "Temperature during the last week of incubation. i. effects on hatching pattern and broiler chicken embryonic organ development". *Poultry Science*, 95(4):956–965, jan 2016a. doi:10.3382/ps/pev447.

C. M. Maatjens, I. A. M. van Roovert-Reijrink, I. van den Anker, B. Engel, C. W. van der Pol, B. Kemp, and H. van den Brand. "Temperature during the last week of incubation. II. effects on first week broiler development and performance". *Poultry Science*, 95(9):2136–2144, apr 2016b. doi:10.3382/ps/pew145.

K. Madsen, H. B. Nielsen, and O. Tingleff. "Methods for non-linear least squares problems (2nd ed.)". Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004.

G. J. Maeda, I. R. Manchester, and D. C. Rye. "Combined ILC and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation". *IEEE Transactions on Control Systems Technology*, 23(5):1754–1769, sep 2015. doi:10.1109/tcst.2014.2382579.

R. O. Manning and R. D. Wyatt. “Effect of cold acclimation on the broiler chicks' resistance to acute aflatoxicosis”. *Poultry Science*, 69(3):388–396, mar 1990. doi:10.3382/ps.0690388.

D. W. Marquardt. “An algorithm for least-squares estimation of nonlinear parameters”. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, jun 1963. doi:10.1137/0111030.

M. A. Márquez-Vera, L. E. Ramos-Velasco, J. Suárez-Cansino, and C. A. Márquez-Vera. “Fuzzy iterative learning control applied in a biological reactor using a reduced number of measures”. *Applied Mathematics and Computation*, 246:608–618, nov 2014. doi:10.1016/j.amc.2014.08.072.

J. D. May and B. D. Lott. “The effect of environmental temperature on growth and feed conversion of broilers to 21 days of age”. *Poultry Science*, 79(5):669–671, may 2000. doi:10.1093/ps/79.5.669.

J. D. May and B. D. Lott. “Relating weight gain and feed:gain of male and female broilers to rearing temperature”. *Poultry Science*, 80(5):581–584, may 2001. doi:10.1093/ps/80.5.581.

R. May, G. Dandy, and H. Maier. “Review of input variable selection methods for artificial neural networks”. In *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech, apr 2011. doi:10.5772/16004.

R. J. May, G. C. Dandy, H. R. Maier, and J. B. Nixon. “Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems”. *Environmental Modelling & Software*, 23(10-11):1289–1299, oct 2008a. doi:10.1016/j.envsoft.2008.03.008.

R. J. May, H. R. Maier, G. C. Dandy, and T. G. Fernando. “Non-linear variable selection for artificial neural networks using partial mutual information”. *Environmental Modelling & Software*, 23(10-11):1312–1326, oct 2008b. doi:10.1016/j.envsoft.2008.03.007.

B. Minne and E. Decuypere. “Effects of late prenatal temperatures on some thermoregulatory aspects in young chickens”. *Archive Experimental Veterinary of Medicine*, 38:374–383, 1984.

O. A. Moldes, J. C. Mejuto, R. Rial-Otero, and J. Simal-Gandara. “A critical review on the applications of artificial neural networks in winemaking technology”. *Critical Reviews in Food Science and Nutrition*, 57:2896 to 2908, 10 2015. doi:10.1080/10408398.2015.1078277.

V. S. Morita, V. R. Almeida, J. B. M. Junior, T. I. Vicentini, H. van den Brand, and I. C. Boleli. “Incubation temperature alters thermal preference and response to heat stress of broiler chickens along the rearing phase”. *Poultry Science*, 95(8): 1795–1804, mar 2016. doi:10.3382/ps/pew071.

N. A. Musharaf and J. D. Latshaw. “Heat increment as affected by protein and amino acid nutrition”. *World's Poultry Science Journal*, 55(03):233–240, sep 1999. doi:10.1079/wps19990017.

I. A. Naas, I. C. L. A. Paz, M. S. Baracho, A. G. Menezes, L. G. F. Bueno, I. C. L. Almeida, and D. J. Moura. “Impact of lameness on broiler well-being”. *The Journal of Applied Poultry Research*, 18(3):432–439, jan 2009. doi:10.3382/japr.2008-00061.

V. V. Nair, H. Dhar, S. Kumar, A. K. Thalla, S. Mukherjee, and J. W. Wong. “Artificial neural network based modeling to evaluate methane yield from biogas in a laboratory-scale anaerobic bioreactor”. *Bioresource Technology*, 217:90–99, oct 2016. doi:10.1016/j.biortech.2016.03.046.

R. Nasimi and R. Irani. “Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks”. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14):1604–1611, may 2014. doi:10.1080/15567036.2011.592903.

National Chicken Council. “Per capita consumption of poultry and livestock, 1965 to estimated 2018”, 2018a. URL: https://www.nationalchickencouncil.org/about-the-industry/statistics/per-capita-consumption-of-poultry-and-livestock-1965-to-estimated-2012-in-pounds/.

National Chicken Council. “U.s. broiler performance”, 2018b. URL: https://www.nationalchickencouncil.org/about-the-industry/statistics/u-s-broiler-performance/.

N. M. Nawi, W. H. Atomi, and M. Rehman. “The effect of data pre-processing on optimized training of artificial neural networks”. *Procedia Technology*, 11:32–39, 2013. doi:10.1016/j.protcy.2013.12.159.

D. P. Neves, T. M. Banhazi, and I. A. Nääs. “Feeding behaviour of broiler chickens: a review on the biomechanical characteristics”. *Revista Brasileira de Ciência Avícola*, 16(2):01–16, jun 2014. doi:10.1590/1516-635x16021-16.

D. P. Neves, S. A. Mehdizadeh, M. Tscharke, I. de Alencar Nääs, and T. M. Banhazi. “Detection of flock movement and behaviour of broiler chickens at different feeders using image analysis”. *Information Processing in Agriculture*, 2(3-4):177–182, oct 2015. doi:10.1016/j.inpa.2015.08.002.

R. C. Newberry, J. R. Hunt, and E. E. Gardiner. “Behaviour of roaster chickens towards an automatic weighing perch”. *British Poultry Science*, 26(2):229–237, apr 1985. doi:10.1080/00071668508416808.

D. Nguyen and B. Widrow. “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights”. In *1990 IJCNN International Joint Conference on Neural Networks*. Institute of Electrical & Electronics Engineers (IEEE), 1990. doi:10.1109/ijcnn.1990.137819.

H. Ninomiya. “A novel quasi-newton-based optimization for neural network training incorporating nesterov's accelerated gradient”. *Nonlinear Theory and Its Applications, IEICE*, 8(4):289–301, 2017. doi:10.1587/nolta.8.289.

D. Nyuiadzi, A. Travel, B. Méda, C. Berri, L. A. Guilloteau, V. Coustham, Y. Wang, J. K. Tona, and A. Collin. “Effect of low incubation temperature and low

ambient temperature until 21 days of age on performance and body temperature in fast-growing chickens". *Poultry Science*, 96(12):4261–4269, oct 2017. doi:10.3382/ps/pex264.

OECD. "OECD-FAO agricultural outlook 2015", jul 2015.

OECD. *OECD-FAO Agricultural Outlook 2016-2025*. OECD Publishing, jul 2016. doi:10.1787/agr_outlook-2016-en.

OECD. *OECD-FAO Agricultural Outlook 2017-2026*. OECD Publishing, jul 2017. doi:10.1787/agr_outlook-2017-en.

OECD. *OECD-FAO Agricultural Outlook 2018-2027*. OECD Publishing, jul 2018a. doi:10.1787/agr_outlook-2018-en.

OECD. "Meat consumption (indicator)". 2018b. doi:10.1787/fa290fd0-en.

A. Olfati, A. Mojtahedin, T. Sadeghi, M. Akbari, and F. Martínez-Pastor. "Comparison of growth performance and immune responses of broiler chicks reared under heat stress, cold stress and thermoneutral conditions". *Spanish Journal of Agricultural Research*, 16(2):e0505, jul 2018. doi:10.5424/sjar/2018162-12753.

A. Overgaard, C. S. Kallesoe, J. D. Bendtsen, and B. K. Nielsen. "Input selection for return temperature estimation in mixing loops using partial mutual information with flow variable delay". In *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, aug 2017. doi:10.1109/ccta.2017.8062650.

D. Panagiotopoulos, C. Orovas, and D. Syndoukas. "A heuristically enhanced gradient approximation (HEGA) algorithm for training neural networks". *Neurocomputing*, 73(7-9):1303–1323, mar 2010. doi:10.1016/j.neucom.2009.12.014.

W. Paszke, E. Rogers, K. Galkowski, and Z. Cai. "Robust finite frequency range iterative learning control design and experimental verification". *Control Engineering Practice*, 21(10):1310–1320, 2013.

H. Peng, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin. "A comparative study on regularization strategies for embedding-based neural networks". In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015. doi:10.18653/v1/d15-1252.

B. Pérez-Sánchez, O. Fontenla-Romero, and B. Guijarro-Berdiñas. "A review of adaptive online learning for artificial neural networks". *Artificial Intelligence Review*, 49:281 to 299, 10 2016. doi:10.1007/s10462-016-9526-2.

M. Petek, G. Sönmez, H. Yildiz, and H. Baspinar. "Effects of different management factors on broiler performance and incidence of tibial dyschondroplasia". *British Poultry Science*, 46(1):16–21, feb 2005. doi:10.1080/00071660400023821.

F. Piekniewski and L. Rybicki. "Visual comparison of performance for different activation functions in MLP networks". *IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, 2004. doi:10.1109/ijcnn.2004.1381133.

Y. Piestun, D. Shinder, M. Ruzal, O. Halevy, J. Brake, and S. Yahav. "Thermal manipulations during broiler embryogenesis: Effect on the acquisition of thermotolerance". *Poultry Science*, 87(8):1516–1525, aug 2008a. doi:10.3382/ps.2008-00030.

Y. Piestun, D. Shinder, M. Ruzal, O. Halevy, and S. Yahav. "The effect of thermal manipulations during the development of the thyroid and adrenal axes on in-hatch and post-hatch thermoregulation". *Journal of Thermal Biology*, 33(7):413–418, oct 2008b. doi:10.1016/j.jtherbio.2008.06.007.

Y. Piestun, M. Harel, M. Barak, S. Yahav, and O. Halevy. "Thermal manipulations in late-term chick embryos have immediate and longer term effects on myoblast proliferation and skeletal muscle hypertrophy". *Journal of Applied Physiology*, 106 (1):233–240, jan 2009. doi:10.1152/japplphysiol.91090.2008.

Y. Piestun, O. Halevy, D. Shinder, M. Ruzal, S. Druyan, and S. Yahav. "Thermal manipulations during broiler embryogenesis improves post-hatch performance under hot conditions". *Journal of Thermal Biology*, 36(7):469–474, oct 2011. doi:10.1016/j.jtherbio.2011.08.003.

Y. Piestun, S. Druyan, J. Brake, and S. Yahav. "Thermal manipulations during broiler incubation alter performance of broilers to 70 days of age". *Poultry Science*, 92(5):1155–1163, apr 2013. doi:10.3382/ps.2012-02609.

D. Prichard and J. Theiler. "Generating surrogate data for time series with several simultaneously measured variables". *Physical Review Letters*, 73(7):951–954, aug 1994. doi:10.1103/physrevlett.73.951.

M. S. Rahman, M. A. H. Pramanik, B. Basak, S. U. Tarafdar, and S. K. Biswas. "Effect of feeding low protein diets on the performance of broiler during hot-humid season". *International Journal of Poultry Science*, 1(123):35–39, dec 2002. doi:10.3923/ijps.2002.35.39.

C. D. Rajamani Doraiswami and M. Stevenson. "Closed loop identification". In *Identification of Physical Systems*, pages 357–378. Wiley-Blackwell, mar 2014. doi:10.1002/9781118536483.ch8.

B. C. Rakshit, M. Ohid Ullah, and D. Mishra. "Thermal manipulation in broilers and layers". *Journal of Applied Quantitative Methods*, 11(2):62 – 71, 2016.

M. Z. Rehman and N. M. Nawi. "The effect of adaptive momentum in improving the accuracy of gradient descent back propagation algorithm on classification problems". In J. Mohamad Zain, W. M. b. Wan Mohd, and E. El-Qawasmeh, editors, *Software Engineering and Computer Systems*, pages 380–390, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-22170-5.

V. Roll, M. Dai PrÃ¡, A. Roll, E. Xavier, P. Rossi, M. Anciuti, and F. Rutz. "Influência da altura de comedouros tubulares no comportamento ingestivo de frangos de corte (effect of tubular feeder height on ingestive behavior of broiler)". *Archivos de Zootecnia*, 59:115 – 122, 03 2010. ISSN 0004-0592. URL: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0004-05922010000100012&nrm=iso.

E. M. Roushdy, A. W. Zaglool, and M. S. El-Tarabany. “Effects of chronic thermal stress on growth performance, carcass traits, antioxidant indices and the expression of HSP70 , growth hormone and superoxide dismutase genes in two broiler strains”. *Journal of Thermal Biology*, 74:337–343, may 2018. doi:10.1016/j.jtherbio.2018.04.009.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. *Nature*, 323:533 to 536, 10 1986. doi:10.1038/323533a0.

M. E. Sabry, S. Yalçin, and G. Turgay-İzzetoğlu. “Effect of breeder age and lighting regimen on growth performance, organ weights, villus development, and bursa of fabricius histological structure in broiler chickens”. *Czech Journal of Animal Science*, 60(No. 3):116–122, jul 2016. doi:10.17221/8076-cjas.

M. I. E. Sabry, S. Yalçın, and G. Turgay-İzzetoğlu. “Interaction between breeder age and hatching time affects intestine development and broiler performance”. *Livestock Science*, 157(2-3):612–617, nov 2013. doi:10.1016/j.livsci.2013.07.012.

M. Sadeghassadi, C. J. Macnab, B. Gopaluni, and D. Westwick. “Application of neural networks for optimal-setpoint design and MPC control in biological wastewater treatment”. *Computers & Chemical Engineering*, 115:150–160, jul 2018. doi:10.1016/j.compchemeng.2018.04.007.

K. M. Sameer Agarwal and Others. “Ceres solver”. `http://ceres-solver.org`, 2015.

A. Sander, D. Skansi, and N. Bolf. “Heat and mass transfer models in convection drying of clay slabs”. *Ceramics International*, 29(6):641–653, jan 2003. doi:10.1016/s0272-8842(02)00212-2.

A. Sharma. “Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — a strategy for system predictor identification”. *Journal of Hydrology*, 239(1-4):232–239, dec 2000. doi:10.1016/s0022-1694(00)00346-2.

D. Shen and Y. Wang. “Survey on stochastic iterative learning control”. *Journal of Process Control*, 24(12):64–77, dec 2014. doi:10.1016/j.jprocont.2014.04.013.

Q. Shen, W. min Shi, X. ping Yang, and B. xian Ye. “Particle swarm algorithm trained neural network for qsar studies of inhibitors of platelet-derived growth factor receptor phosphorylation”. *European Journal of Pharmaceutical Sciences*, 28:369 to 376, 08 2006. doi:10.1016/j.ejps.2006.04.001.

J. Sohl-Dickstein, B. Poole, and S. Ganguli. “Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods”. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 604–612, Bejing, China, 22–24 Jun 2014. PMLR. URL: `http://proceedings.mlr.press/v32/sohl-dicksteinb14.html`.

K. Stacey, D. Parsons, A. Frost, C. Fisher, D. Filmer, and A. Fothergill. “An automatic growth and nutrition control system for broiler production”. *Biosystems Engineering*, 89(3):363–371, nov 2004. doi:10.1016/j.biosystemseng.2004.07.006.

H. sung Ahn, K. Moore, and Y. Chen. “LMI approach to iterative learning control design”. jul 2006a. doi:10.1109/smcals.2006.250694.

H. sung Ahn, K. Moore, and Y. Chen. “LMI approach to iterative learning control design”. jul 2006b. doi:10.1109/smcals.2006.250694.

S. Syafwan, G. J. D. Wermink, R. P. Kwakkel, and M. W. A. Verstegen. “Dietary self-selection by broilers at normal and high temperature changes feed intake behavior, nutrient intake, and performance”. *Poultry Science*, 91(3):537–549, feb 2012. doi:10.3382/ps.2011-01559.

C. W. Tallentire, I. Leinonen, and I. Kyriazakis. “Publisher correction: Artificial selection for improved energy efficiency is reaching its limits in broiler chickens”. *Scientific Reports*, 8(1), mar 2018. doi:10.1038/s41598-018-23133-8.

G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein. “Training neural networks without gradients: A scalable admm approach”. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2722–2731, New York, New York, USA, 20–22 Jun 2016. PMLR. URL: http://proceedings.mlr.press/v48/taylor16.html.

A. C. Tsoi. *Recurrent neural network architectures: An overview*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-69752-7. doi:10.1007/BFb0053993.

A. C. Tsoi and A. Back. “Discrete time recurrent neural network architectures: A unifying review”. *Neurocomputing*, 15:183 to 223, 06 1997. doi:10.1016/S0925-2312(97)00161-6.

E. Tullo, I. Fontana, A. P. Fernandez, E. Vranken, T. Norton, D. Berckmans, and M. Guarino. “Association between environmental predisposing risk factors and leg disorders in broiler chickens1, 2”. *Journal of Animal Science*, 95(4):1512–1520, apr 2017. doi:10.2527/jas.2016.1257.

M. Uchiyama. “Formulation of high-speed motion pattern of mechanical arm by trial”. *Trans. Soc. Instrum. Control Eng.*, 14(6):706–712, 1978. URL: http://ci.nii.ac.jp/naid/10026860506/en/.

A. M. Ulmer-Franco, G. M. Fasenko, and E. E. O. Christopher. “Hatching egg characteristics, chick quality, and broiler performance at 2 breeder flock ages and from 3 egg weights”. *Poultry Science*, 89(12):2735–2742, dec 2010. doi:10.3382/ps.2009-00403.

A. M. Ulmer-Franco, G. Cherian, N. Quezada, G. M. Fasenko, and L. M. McMullen. “Hatching egg and newly hatched chick yolk sac total IgY content at 3 broiler breeder flock ages”. *Poultry Science*, 91(3):758–764, feb 2012. doi:10.3382/ps.2011-01757.

K. P. Unnikrishnan and K. P. Venugopal. “Alopex: A correlation-based learning algorithm for feedforward and recurrent neural networks”. *Neural Computation*, 6 (3):469–490, may 1994. doi:10.1162/neco.1994.6.3.469.

J. van Zundert, J. Bolder, S. Koekebakker, and T. Oomen. "Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer". *Mechatronics*, 38:76–90, sep 2016. doi:10.1016/j.mechatronics.2016.07.001.

S. L. Vieira and C. R. Angel. "Optimizing broiler performance using different amino acid density diets: What are the limits?". *The Journal of Applied Poultry Research*, 21(1):149–155, mar 2012. doi:10.3382/japr.2011-00476.

N. X. Vinh, S. Zhou, J. Chan, and J. Bailey. "Can high-order dependencies improve mutual information based feature selection?". *Pattern Recognition*, 53:46–58, may 2016. doi:10.1016/j.patcog.2015.11.007.

H. Wang, X. Yan, H. Chen, C. Chen, and M. Guo. "Chlorophyll-a predicting model based on dynamic neural network". *Applied Artificial Intelligence*, 29(10):962–978, nov 2015. doi:10.1080/08839514.2015.1097142.

H. Wang, R. Feng, Z.-F. Han, and C.-S. Leung. "Admm-based algorithm for training fault tolerant rbf networks and selecting centers". *IEEE Transactions on Neural Networks and Learning Systems*, 29:3870 to 3878, 08 2018. doi:10.1109/TNNLS.2017.2731319.

L. Wang, Y. Yan, X. Wang, and T. Wang. "Input variable selection for data-driven models of coriolis flowmeters for two-phase flow measurement". *Measurement Science and Technology*, 28(3), 2017. URL: `http://stacks.iop.org/0957-0233/28/i=3/a=035305`.

Y. Wang, F. Gao, and F. J. Doyle. "Survey on iterative learning control, repetitive control, and run-to-run control". *Journal of Process Control*, 19(10):1589–1600, dec 2009. doi:10.1016/j.jprocont.2009.09.006.

C. Wathes, H. Kristensen, J.-M. Aerts, and D. Berckmans. "Is precision livestock farming an engineer's daydream or nightmare, an animal's friend or foe, and a farmer's panacea or pitfall?". *Computers and Electronics in Agriculture*, 64(1): 2–10, nov 2008. doi:10.1016/j.compag.2008.05.005.

D. R. Webb. "Thermal tolerance of avian embryos: A review". *The Condor*, 89(4): 874, nov 1987. doi:10.2307/1368537.

C. A. Weeks. "Poultry handling and transport.". In *Livestock handling and transport*, volume 4, chapter 20, pages 378–398. CABI, 2014. ISBN 9781780643212. doi:10.1079/9781780643212.0378.

L. D. Wet, E. Vranken, A. Chedad, J.-M. Aerts, J. Ceunen, and D. Berckmans. "Computer-assisted image analysis to quantify daily growth rates of broiler chickens". *British Poultry Science*, 44(4):524–532, sep 2003. doi:10.1080/00071660310001616192.

S. S. Wolfgang Härdle, Marlene Müller and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer, 2004.

J.-X. Xu. "A survey on iterative learning control for nonlinear systems". *International Journal of Control*, 84(7):1275–1294, jul 2011. doi:10.1080/00207179.2011.574236.

Z. Xu, J. Zhao, Y. Yang, Z. Shao, and F. Gao. "Optimal iterative learning control based on a time-parametrized linear time-varying model for batch processes". *Industrial & Engineering Chemistry Research*, 52(18):6182–6192, may 2013. doi:10.1021/ie302561t.

S. Yalçin, S. Özkan, L. Türkmut, and P. B. Siegel. "Responses to heat stress in commercial and local broiler stocks. 1. performance traits". *British Poultry Science*, 42(2):149–152, may 2001. doi:10.1080/00071660120048375.

H. Yang, H. Xing, Z. Wang, J. Xia, Y. Wan, B. Hou, and J. Zhang. "Effects of intermittent lighting on broiler growth performance, slaughter performance, serum biochemical parameters and tibia parameters". *Italian Journal of Animal Science*, 14(4):4143, jan 2015. doi:10.4081/ijas.2015.4143.

Y. Yang, C. Pan, R. Zhong, and J. Pan. "Artificial light and biological responses of broiler chickens: dose-response". *Journal of Animal Science*, 96(1):98–107, jan 2018. doi:10.1093/jas/skx044.

H. Yu and B. M. Wilamowski. "Neural network training with second order algorithms", 2018.

V. D. Yunianto, K. Hayashit, S. Kaiwda, A. Ohtsuka, and Y. Tomita. "Effect of environmental temperature on muscle protein turnover and heat production in tube-fed broiler chickens". *British Journal of Nutrition*, 77(06):897, jun 1997. doi:10.1079/bjn19970088.

Q. U. Zaman, T. Mushtaq, H. Nawaz, M. A. Mirza, S. Mahmood, T. Ahmad, M. E. Babar, and M. M. H. Mushtaq. "Effect of varying dietary energy and protein on broiler performance in hot climate". *Animal Feed Science and Technology*, 146(3-4): 302–312, oct 2008. doi:10.1016/j.anifeedsci.2008.01.006.

J. Zhang. "Batch-to-batch optimal control of a batch polymerisation process based on stacked neural network models". *Chemical Engineering Science*, 63(5):1273–1281, mar 2008. doi:10.1016/j.ces.2007.07.047.

M. J. Zuidhof, B. L. Schneider, V. L. Carney, D. R. Korver, and F. E. Robinson. "Growth, efficiency, and yield of commercial broilers from 1957, 1978, and 20051". *Poultry Science*, 93(12):2970–2982, nov 2014. doi:10.3382/ps.2014-04291.

I. Zulkifli, A. F. Akmal, A. F. Soleimani, M. A. Hossain, and E. A. Awad. "Effects of low-protein diets on acute phase proteins and heat shock protein 70 responses, and growth performance in broiler chickens under heat stress condition". *Poultry Science*, 97(4):1306–1314, mar 2018. doi:10.3382/ps/pex436.

Part II

Papers

Paper A

Data Driven Broiler Weight Forecasting using Dynamic Neural Network Models

Simon V. Johansen[a,b], Jan D. Bendtsen[b], Martin R.-Jensen[a] and Jesper Mogensen[b]

[a]SKOV A/S, Hedelund 4, Glyngøre, Denmark
[b]Department of Control and Automation, Aalborg University, Denmark

Abstract—*In this article, the dynamic influence of environmental broiler house conditions and broiler growth is investigated. Dynamic neural network forecasting models have been trained on farm-scale broiler batch production data from 12 batches from the same house. The model forecasts future broiler weight and uses environmental conditions such as heating, ventilation, and temperature along with broiler behavior such as feed and water consumption. Training data and forecasting data is analyzed to explain when the model might fail at generalizing. We present ensemble broiler weight forecasts to day 7, 14, 21, 28 and 34 from all preceding days and provide our interpretation of the results. Results indicate that the dynamic interconnection between environmental conditions and broiler growth can be captured by the model. Furthermore, we found that a comparable forecast can be obtained by using input data from the previous batch as a substitute for future input data.*

Keywords—*Biosystems; Neural networks; Agricultural engineering*

Published in *Proceedings of World Congress of the International Federation of Automatic Control*, 2017.

Contents

1 Introduction

Due to the growing middle class in many developing countries, the increase in meat consumption is predicted to continue rising [OECD, 2015]. The highest growth in meat production is foreseen in poultry meat, of which we expect broiler meat to represent the majority. The reason being that broiler production gives the highest yield per feed unit among land animals, making chicken meat a relatively inexpensive source of animal protein. The poultry industry on a world scale is predicted to steadily increase, from an average of 107.6 billion kg of meat in 2012-2014 to 134.8 in 2024 [OECD, 2015, pp. 136] – a significant predicted increase of 24%.

Available scientific literature on dynamic broiler models focuses mainly on active broiler weight control by regulating feed uptake and composition, which traditionally favors simplistic models [Wathes et al., 2008], [Aerts et al., 2003]. Feed is generally considered the biggest expense in broiler production, and correct climate control is known to yield superior feed utilization for broiler growth. However, no scientific literature has been found that studies the complex dynamic interconnection between broiler weight and broiler house environmental conditions, and will thus be the subject of this work. Such a model has the potential of allowing active broiler weight control by regulating the broiler environment, which is nonexistent in industry today. Especially considering that most broiler houses use ad libitum feeding regimes, which excludes the regulation of feed uptake.

Traditionally, empirically motivated nonlinear growth curve models have been used to determine evolution of broiler weight. It has been extensively studied in scientific literature such as [Aggrey, 2002], [Ahmadi and Mottaghitalab, 2007] and [Hasan Eleroğlu and Duman, 2014]. A growth curve in this context is a static curve fitted to old broiler weight data. It has a fixed structure with few parameters that offers biologically intuitive interpretations. Common models include the Richerds Model and Gompertz-Laird Model that are described by 4 and 3 parameters respectively, where the parameters have biologically intuitive interpretations – such as time and size of maximum growth rate [Aggrey, 2002]. In [Lopes et al., 2008], the relation between broiler house environment and production performance was investigated using a neural network. However, it is not clear how to extract temporal performance (i.e. growth prediction) from this type of model.

Dynamic broiler growth models are an extension of growth curves and has primarily been developed for control synthesis in scientific literature. In [Aerts et al., 2003] a time-variant online parameter estimation of a dynamic model was successfully applied to predict future weight up to 7 days without a priori information. The model use feed intake as input and weight as an output and was used for model predictive control (MPC) in [Cangar et al., 2007]. This control algorithm was tested in a laboratory setting with a stocking density of 5.3 birds/m^2 and 20 birds/m^2, the later to emulate farm scale density. The mean relative weight control error was 2.7% and 7.3% for the low and high-

density experiments respectively – suggesting that farm scale broiler production is harder to both predict and control.

A similar result was obtained in [Demmers et al., 2010], where a small differential recurrent neural network was used to model the feed quantity and control the broiler weight using nonlinear MPC. In [Stacey et al., 2004] a dynamic broiler weight model was developed and used information about feed uptake and composition of two feed types with known nutritional value. The model was successfully used to control broiler weight, it was tested on farm scale with 30,000-40,000 broilers per house and achieved results comparable to that of a stockman. However, these studies do not take environmental conditions into account.

In this paper we use neural network class models for forecasting, which is a soft computing technique – a group of inexact methods that is capable of dealing with uncertainty. More specifically, we will be using dynamic neural networks, which has been successfully applied to model complex biological processes. Recent applications include algae growth prediction in a laboratory setting [Wang et al., 2015], prediction of bioethanol production in a bioreactor [Grahovac et al., 2016], bioreactor prediction [Nair et al., 2016], yeast fermentation modeling in a bioreactor [Nasimi and Irani, 2014] and state estimation in a continuous bio reactor [Hernandez et al., 2013].

In the present effort, we present a data-driven dynamic neural network broiler batch forecasting model. In particular, we show that dynamic interaction between environmental conditions and chicken behavior in ad libitum feed broilers is present and can be captured by a model. We do this in a data driven framework on real farm scale production data from a state of the art broiler house.

We have to emphasize that state of art broiler production is mainly empirically driven. For this reason, we use a broiler expert employed by SKOV A/S to interpret both the input variables and the results presented in this work.

The remainder of the article is structured as follows. In Section 2 the model, training and validation method is described. In Section 3 the data collection and analysis, model configuration and forecasting results are described. In Section 4.1 we discuss the obtained experimental results followed by concluding remarks in Section 5.

2 Model, Training and Validation Method

2.1 Model

Since the multilayer perceptron (MLP) model and its variations have been extensively researched in scientific literature we will only give a brief introduction and describe how we apply it to our specific problem. For a thorough introduction to the subject we redirect the reader to [Du and Swamy, 2014] and [Haykin, 1994].

The particular type of Dynamic Neural Network (DNN) model we use in this work can be classified as a discrete nonlinear ARX model of the form

$$\hat{y}[k+1 \mid \mathcal{W}] = \mathcal{N}\big(\hat{Y}[k], U[k+1] \mid \mathcal{W}\big) \tag{1}$$

with

$$U[k+1] = \begin{bmatrix} u[k-n_1+1]^T & \cdots & u[k-n_i+1]^T \end{bmatrix}^T \text{ and} \tag{2a}$$

$$\hat{Y}[k] = \begin{bmatrix} \hat{y}[k-m_1 \mid \mathcal{W}]^T & \cdots & \hat{y}[k-m_j \mid \mathcal{W}] \end{bmatrix}^T, \tag{2b}$$

where $\mathcal{N}$ is a MLP model, $U[k]$ is delayed values of the input vector $u[k] \in \mathbb{R}^N$ corresponding to the i elements of $n = \{n_1, \cdots, n_i\}$, $\hat{Y}[k]$ is delayed values of the previous output vector $\hat{y}[k] \in \mathbb{R}^M$ corresponding to the j elements of $m = \{m_1, \cdots, m_j\}$ and $\mathcal{W}$ is an abstract representation of all the model weights in a vector. This particular structure has been adopted to accommodate potentially long propagation delays, while still aiming to keep the number of weights relatively low.

The DNN model $\mathcal{N}$ is selected with one hidden layer with hyperbolic tangent activation function in the hidden layer and linear activation function in the output layer. It can be shown that $\mathcal{N}$ is an universal function approximator, meaning that it has the capacity for approximating any system to any accuracy [Du and Swamy, 2014, chap. 4.2]. In matrix-vector representation, (1) equals

$$\hat{y}[k+1 \mid \mathcal{W}] = W^{\mathrm{o}} \tanh(\mathcal{X}) + \theta^{\mathrm{o}} \tag{3}$$

with

$$\mathcal{X} = \sum_{a=1}^{j} W^{\mathrm{h}}_{y,a} \hat{y}[k-m_a+1 \mid \mathcal{W}] + \sum_{b=1}^{i} W^{\mathrm{h}}_{u,b} u[k-n_b+1] + \theta^{\mathrm{h}},$$

where W^{o} is the output weights, $W^{\mathrm{h}}_{y,a}$ is the delayed output weights, W^{h}_{b} is the delayed input weights, θ^{h} is the hidden layer bias and θ^{o} is the output bias. A visual representation of (3) is depicted on Figure 1.

2.2 Model Validation

We partition the available batches into three categories, namely training, prediction, and evaluation. In ascending order from the latest batch is the evaluation batch, prediction batch, and the remaining batches are training batches. The training batches are used to find the network weights $\mathcal{W}$ through training. The latest training batch is used in an early stopping setting to avoid over-fitting by selecting the model that generalizes best to this batch.

We generate 64 models with randomly initialized weights and train each one individually on the same training data. We then use the ensemble mean and standard deviation of all the individual model forecast runs to represent the "true" model forecasting output in open loop from day 7 to 34. We will

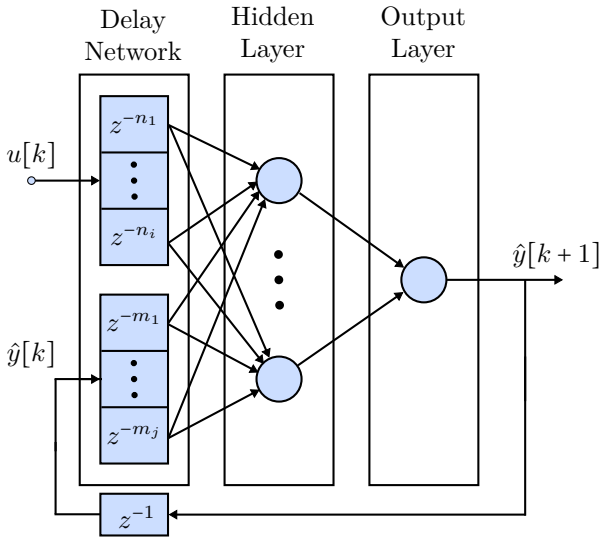


Figure 1: Visual representation of (3) with one input and one output for simplicity. Neurons are represented with blue circles, which contain an activation function and bias each. Note that all incoming signals to a neuron are multiplied by a weight. The operator z^{-a} produce a delay of a samples.

investigate the impact of using old input data from the prediction batch to represent the best guess for future input $U[k]$ in the evaluation batch. To demonstrate the model's ability to forecast broiler weight throughout the batch, we forecast the weight from all preceding samples to day 7, 14, 21, 28 and 34 – denoted *weight on day forecasting*.

2.3 Model Training

The dynamic neural network is trained by minimizing the cost function

$$\underbrace{\frac{1}{B}\sum_{b=1}^{B}\sum_{k=\kappa}^{K_b}\frac{\|y[k\mid b]-\hat{y}[k\mid \kappa,b,\mathcal{W}]\|_2^2}{N_y(K_b-\kappa+1)}}_{E_D}+\bar{\alpha}\underbrace{\|\mathcal{W}\|_2^2}_{E_\mathcal{W}}, \tag{4}$$

where B is the number of training batches, κ is the index of the first output, and K_b is the number of samples in batch b. The target output at sample k given batch b is denoted $y[k \mid b]$, while the predicted network output at sample k, initialized at sample κ, using future input values from batch b with model weights $\mathcal{W}$, is denoted $\hat{y}[k \mid \kappa, b, \mathcal{W}]$. In this context, initialization refers to the initial values of (2). $\|\cdot\|_2$ is the vector 2-norm.

Both the inputs and outputs are normalized to a mean of 0 and standard deviation of 1 during training. The network weights $\mathcal{W}$ are initialized using the Nguyen-Widrow algorithm as explained in [Nguyen and Widrow, 1990].

The last term is a scalar regularization term punishing the size of the $N_{\mathcal{W}}$ system weights $\mathcal{W}$, where $\bar{\alpha}$ is the regularization weight. The regularization weight $\bar{\alpha}$ is determined iteratively through Bayesian Regulation as described in [Burden and Winkler, 2008]. We normalize the regularization weight to make the cost functions comparable between runs, and is calculated according to $\bar{\alpha} = \alpha/\beta$ with:

$$\alpha = \frac{\gamma}{2E_{\mathcal{W}}} \qquad \beta = \frac{N_D - \gamma}{2E_D} \qquad \gamma = N_{\mathcal{W}} - \alpha \operatorname{trace}\left(G^{-1}\right) \tag{5}$$

Where $E_{\mathcal{W}}$ and E_D originate from (4) and γ, $0 \leq \gamma \leq N_{\mathcal{W}}$, is a measure of how many of the $N_{\mathcal{W}}$ parameters are used. Lastly, G is the hessian matrix of the joint cost $\beta E_D + \alpha E_W$ according to (6a), where I is the identity matrix of appropriate dimensions. It is partially approximated through the jacobian J_D of E_D in (6b), similar to the Levenberg Marquard algorithm. The scalar constants α and β are iteratively updated between training epochs, with initial conditions $\alpha = 0$ and $\beta = 1$.

$$G = \frac{d^2(\beta E_D + \alpha E_{\mathcal{W}})}{d\mathcal{W}\, d\mathcal{W}^T} = \frac{d^2(\beta E_D)}{d\mathcal{W}\, d\mathcal{W}^T} + 2\alpha I \tag{6a}$$

$$\approx J_D^T J_D + 2\alpha I \text{ with } J_D = \beta \frac{dE_D}{d\mathcal{W}^T} \tag{6b}$$

The cost function (4) is minimized using the Levenberg-Marquardt optimization algorithm by means of the Ceres Solver library [Sameer Agarwal and Others, 2015]. It is an algorithm that efficiently solves large scale least square problems.

3 Experiment

This work is based on data gathered from a ≈20 year old state of the art broiler house located in the northern Denmark. We use data from 12 batches, collected over a period of 19 months. Each batch contains roughly 40,000 broilers.

3.1 Data Analysis

The inputs to the model consist of the available environmental variables: the measured temperature, humidity and CO_2, and the references of the light intensity, ventilation level, and heating level. The model outputs consist of the available broiler behavior indicators: the measured weight along with feed and water consumption per bird.

We intentionally distinguish between reference, demand and measured variables. Reference variables are independent, demand variables are determined by a deterministic entity like a controller outside the model and measured variables are dependent on the model process.

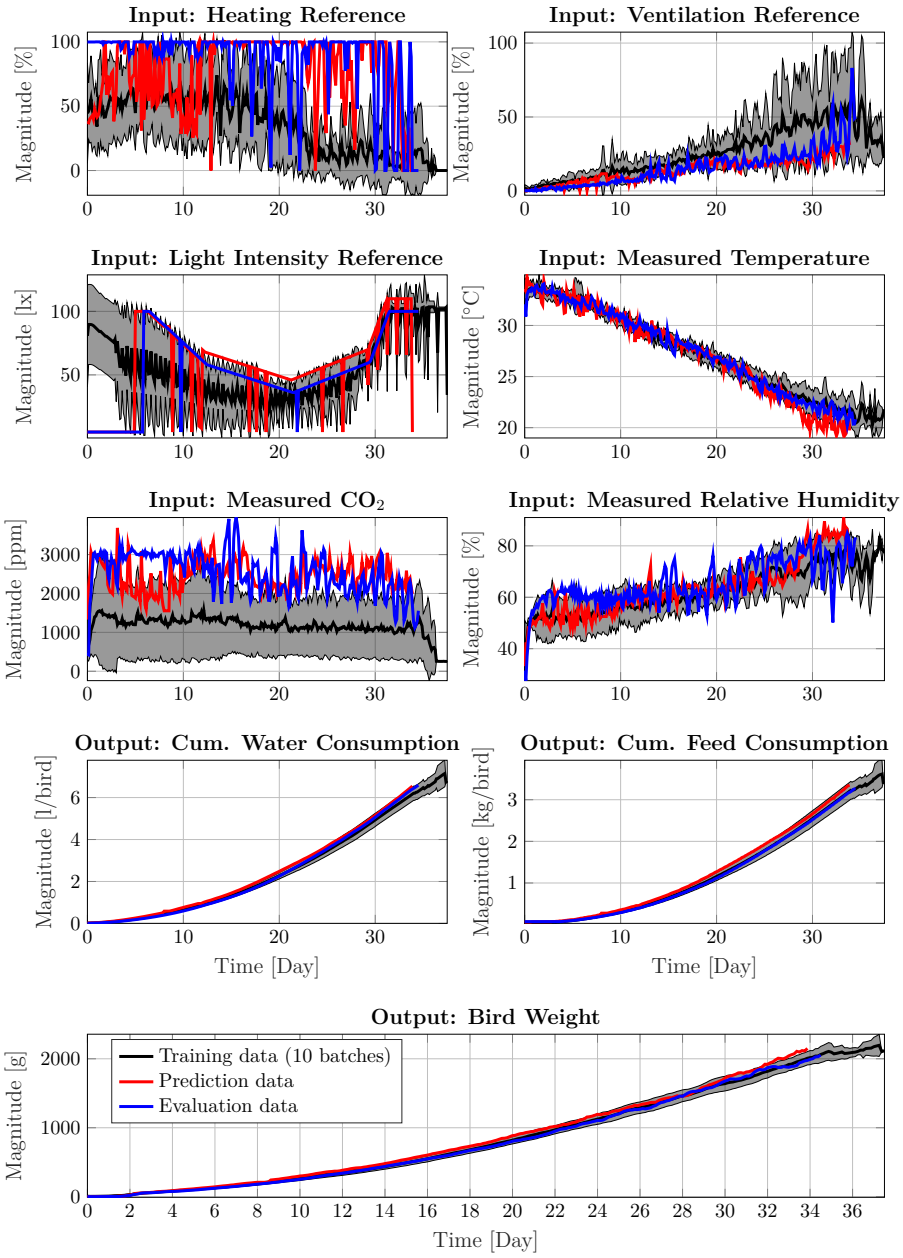


Figure 2: The inputs and output data from the available batches. The black shaded area denotes one standard deviation from the mean value.

To elaborate, humidity, temperature, and CO_2 measurements are measured during closed loop operation and is controlled through both the ventilation and heating demands. For this reason, these variables are not independent, which has been known to cause problems in closed loop identification [Rajamani Doraiswami and Stevenson, 2014]. We do not consider this an issue as all closed loop controlled variables are considered inputs to the model and therefore no deterministic output feedback is present. Keep in mind that reference inputs for controlled variables is determined by the broiler farmer, whom we consider a non-deterministic entity, making it an open loop identification problem with restricted input space. Hence, it still remains to be confirmed if enough excitation of the correlated input variables are present to accurately represent the desired operational input space.

In order to understand the results we give our interpretation of the input and output variables, which are depicted on Figure 2. For this reason, we emphasize the difference between what the model "knows" through training data, and the "unknown" validation and prediction data, which can help explain if the model has difficulties at generalizing. We also point out known sources of error that can influence the model's forecasting ability.

Input variables

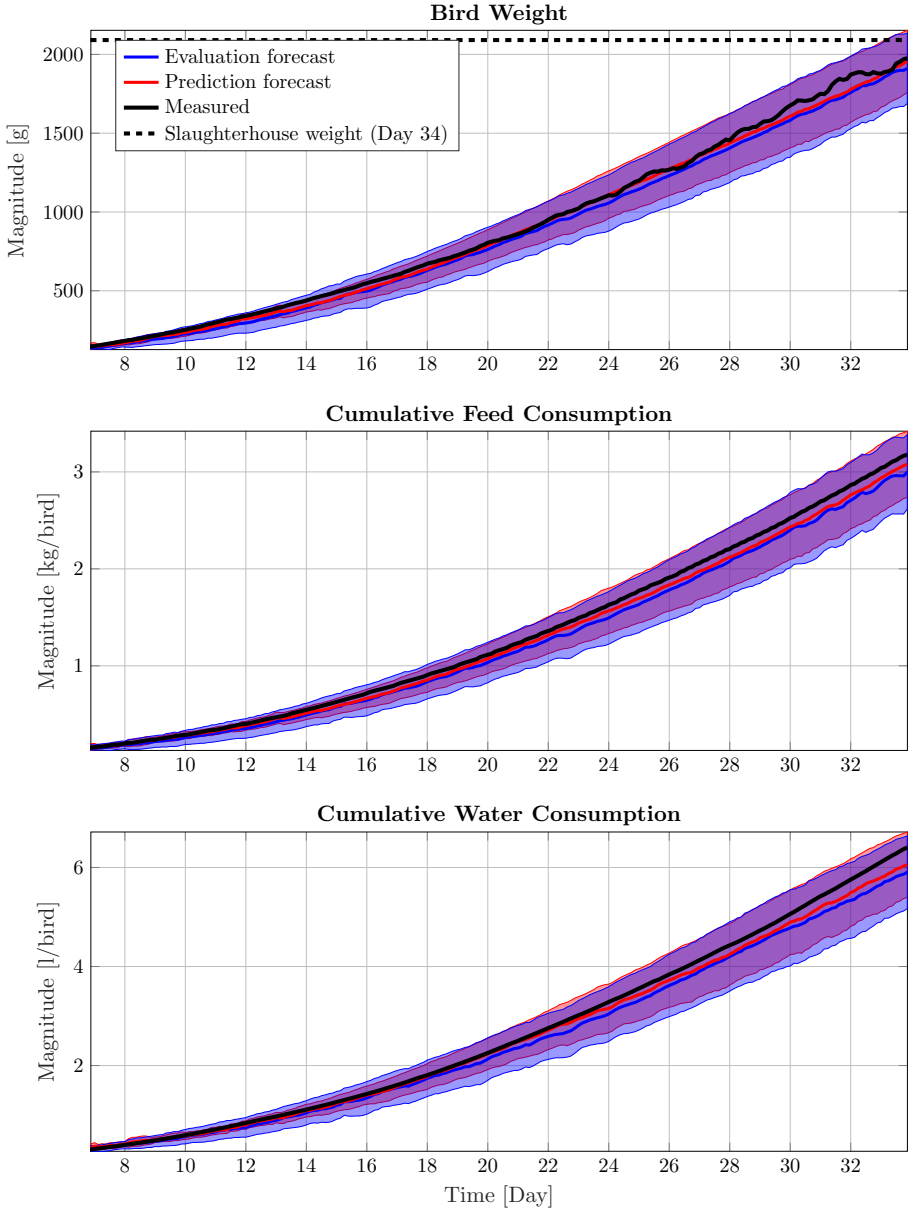
First we give an example of the aforementioned correlated input variables, which is most clearly illustrated on the evaluation batch. Consequently, the ventilation demand is lower and both heating demand and measured CO_2 is higher throughout the batch. As this can be considered significantly different from the training data this will test the models ability to generalize.

As an extension of this, we note that the ventilation demand is negatively correlated with both the measured CO_2 and positively correlated with the measured temperature. Furthermore, the measured humidity appears correlated with the ventilation demand, but this is in fact caused by a larger physical organic mass in the broiler house as the broilers grow throughout the batch.

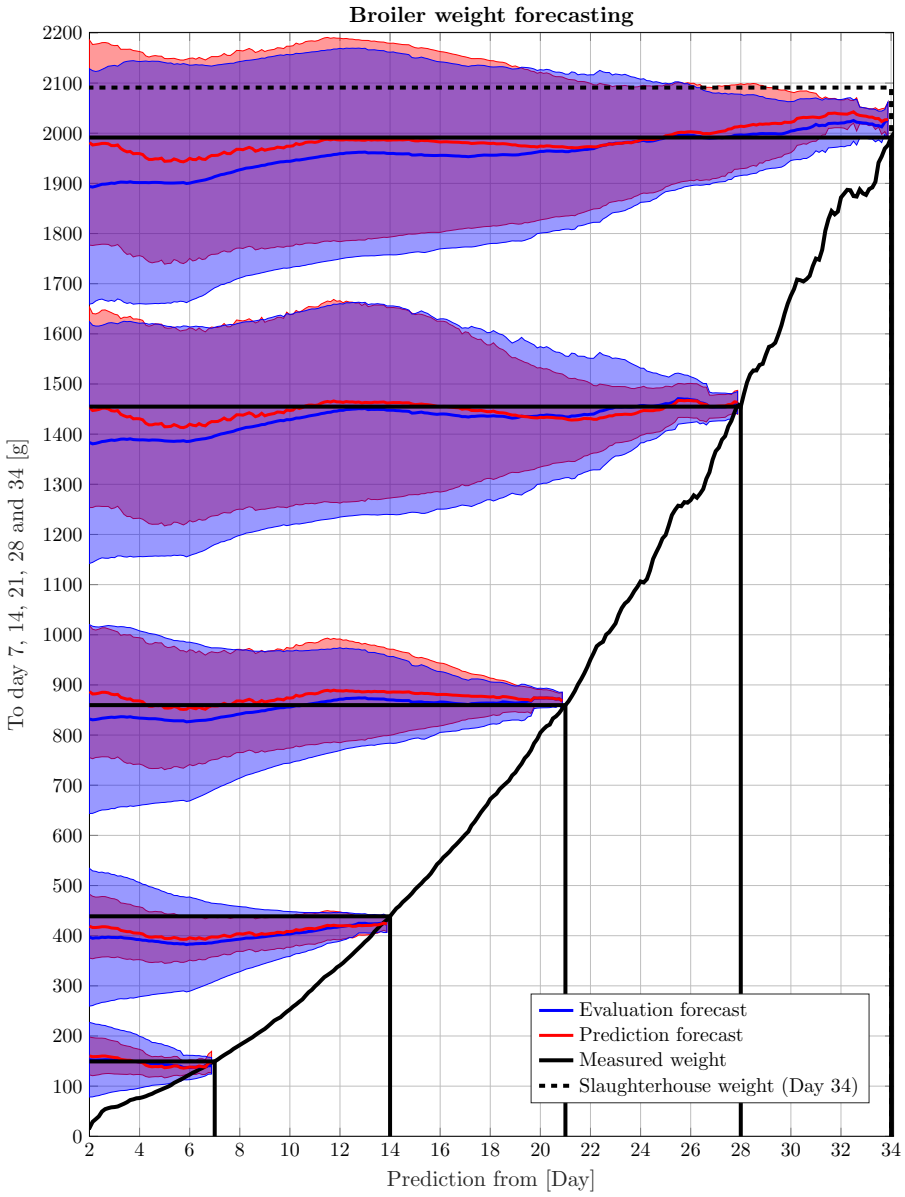
We note a low degree of variation in temperature across all available batches, which we attribute to it being a controlled variable with tighter margins than e.g. CO_2. The prediction batch has a lower than average temperature near the batch end, and for this reason we expect to see deviations if the model fails at generalizing. Consequently the model is only valid for tightly temperature controlled broiler houses. Lastly, we note that the light intensity reference is quite spiky, which is caused by manual overriding by the farmer.

Output variables

First off, broiler weight is highly positively correlated with both the cumulative feed and water consumption, which makes sense as the two are necessary for both survival and growth of the broilers. We note that bird weight has been reported by our customers to be highly predictable from feed and water



(a) Example of forecasting from day 7 to 34.



(b) Weight on day forecasting to day 7, 14, 21, 28 and 34.

Figure 3: The precision of the individual models is illustrated through both the ensemble mean and standard deviation. The solid blue evaluation forecasts are the ensemble mean generated with models initialized with evaluation batch data and future inputs from the evaluation batch. The solid red prediction forecasts are generated with the same procedure, but with future inputs from the prediction batch. To be specific, the blue traces uses future input data from the current batch, where the red traces uses future input data from a previous batch. The blue and red shaded areas are the ensemble standard deviation.

consumption, which can be illustrated by the fact that the three have similar variations and behavior. Furthermore, it is unknown if and when the farmer has changed feed type for any of the batches.

It is a common problem within broiler production that the measured weight is negatively biased from day 15 and onwards. We believe that this among others is due to the broilers outgrowing their skeletal capacity, resulting in a reluctance for the heavier broilers to jump up on the weight pads placed in the broiler house. This has been corrected by multiplying a manually configured and time varying "behavior" constant with the measured weight. We only have the corrected weight which naturally leads to some uncertainty. The projected slaughterhouse weight on day 34 are 2138g and 2091g for the prediction and evaluation batch respectively – a difference of 47g. The weighing pad measured a weight of 2007g and 2140g on this day, resulting in a difference of -84g and 2g compared to the slaughterhouse weight, respectively.

3.2 Model Configuration and Results

For this work we use $h = 7$ hidden neurons with input delays of $n \in \{0, 1, 2\}$ samples and output delays of $m \in \{0, 1, 9\}$ samples in (1), resulting in a model with 220 free parameters. We find that the output delay of ≈ 1 day has a stabilizing effect on the forecasting. We have good experience with a sampling interval of $T_s = 3$ hours and first output training sample $\kappa = 10$ (day 1.25) in (4).

On Figure 3a a forecasting example from day 7 to 34 is presented, while weight on day forecasting on day 7, 14, 21, 28 and 34 from all preceding samples that is depicted on Figure 3b.

4 Discussion

4.1 Forecasting example

In this subsection we discuss Figure 3a. We see that both prediction and evaluation forecasts produce good overall forecasting capabilities, in the sense that it tracks the overall target weight throughout the batch. Comparing the use of prediction and evaluation forecast, we see a slightly higher mean weight of 45g on day 34 for the prediction forecast. This is close to the slaughterhouse weight difference of 47g on this day, despite the weight being ≈ 100g lower ($\approx$9% of the slaughter weight). This indicates that the model captures at least some of the dynamic interconnection between the two forecasts, as the climate conditions are the only difference.

The model uses future prediction of the cumulative feed and water consumption to support the weight forecast. We see that the model underestimates both the feed and water consumption at day 34 by $\approx$6% and $\approx$8% respectively for the evaluation forecast, which can explain part of the $\approx$9% underestimated

weight. It supports that broiler weight can be inferred from the cumulative feed and water consumption, as these figures are quite close to the underestimated slaughter weight.

4.2 Weight on day forecasting

In this subsection we discuss Figure 3b. Again, we see that both prediction and evaluation weight on day forecasts produce good overall forecasting capabilities, in the sense that the forecast is close to the measured weight for the target days.

Common for the weight on day forecasts is that the ensemble standard deviation tends to decrease over time – which is expected as the forecasting horizon diminish. Furthermore, keep in mind that differences between the two forecasts are caused by different future inputs, as they are initialized with the same data – as seen around day 13, where the humidity and CO_2 is exited in the evaluation batch. If both forecasts react, it is caused by the model initialization with the evaluation data – as observed around day 31, where the weight measurement plateaus.

The weight on day 34 prediction forecast tends to be higher than the prediction forecast, which is also the case for the two batches. This difference diminishes gradually from day 7 and onwards, which the model attributes to the overall climate differences in the respective batches. We see that the measured weight on day 34 differs significantly from the projected slaughterhouse weight. The model appears to be able to pick this up just before the broiler weight plateaus at around day 31. This suggests that the model has learned the weighing pad behavior from the training data, and not the "actual" broiler weight.

Only the ensemble standard deviation around day 14 prediction is consistently deviating from the weight measurement, where the others tend to increase throughout the batch. Furthermore, models simulated from around day 8-15 appear to increase the forecasting ensemble standard deviation for both models, indicating that an unfamiliar combination of weight, feed and water consumption is observed in the evaluation data. Alternatively, both the prediction and evaluation batch contains different CO_2 from the training data for this time interval. Regardless, it indicates that the model is having difficulties at generalizing correctly for this interval in particular.

Comparing the evaluation and prediction forecast, we see that they resemble each other well in terms of ensemble standard deviation as the shaded area is similar for the two forecasts, indicating that input data from the previous batch can be used to represent the ensemble standard deviation of the current batch despite different performance.

5 Concluding Remarks

In this present work DNN forecasting models have been trained on farm scale broiler batch production data from 12 batches. We have produced and interpreted a forecast from day 7 to 34 and a weight on day forecast to day 7, 14, 21, 28 and 34 from all preceding days along with displaying the prediction ensemble standard deviation. We found an overall good agreement between measured broiler weight and the weight forecasts, but limited to the measured weight, which is known to be negatively biased onwards of day 15. Most importantly, we found that a dynamic interconnection between environmental broiler house conditions and broiler weight is present, and that it can be captured at least partially by the developed forecasting model. We analyzed the training data and forecasting data to explain the underlying reasons for some of these deviations. Additionally, we found that environmental input data from the previous batch can represent the ensemble standard deviation of the current batch, and is considered a reasonable substitute for future environmental input data.

Future work includes: Investigating state estimation as an alternative to reinitializing the model at every time step in the weight on day forecasting. Establishing a measure of how much the model has to generalize from known training data to forecasting data, as this was an intricate part of evaluating the quality of this work. Determine the optimal number of neurons, input/output delays and input/output variables.

SKOV A/S has submitted a patent application covering the presented concept and intends on implementing the developed forecasting algorithm in existing equipment.

Acknowledgements

The authors would like to thank broiler manager Leif Barsballe for permitting access to the data used in this work, and broiler expert Martin Rishøj at SKOV A/S for assisting in the analysis of the results.

References

J. M. Aerts, M. Lippens, G. De Groote, J. Buyse, E. Decuypere, E. Vranken, and D. Berckmans. “Recursive prediction of broiler growth response to feed intake by using a time-variant parameter estimation method”. *Poultry Science*, 82:40–49, 2003. ISSN 0032-5791.

S. Aggrey. “Comparison of three nonlinear and spline regression models for describing chicken growth curves”. *Poultry Science*, 81(12):1782–1788, dec 2002. doi:10.1093/ps/81.12.1782.

H. Ahmadi and M. Mottaghitalab. “Hyperbolastic models as a new powerful tool to describe broiler growth kinetics”. *Poultry Science*, 86(11):2461–2465, nov 2007. doi:10.3382/ps.2007-00086.

F. Burden and D. Winkler. “Bayesian regularization of neural networks”. In *Methods in Molecular Biology*, pages 23–42. Springer Science + Business Media, 2008. doi:10.1007/978-1-60327-101-1_3.

O. Cangar, J.-M. Aerts, E. Vranken, and D. Berckmans. “Online growth control as an advance in broiler farm management”. *Poultry Science*, 86:439–443, 2007. ISSN 0032-5791.

T. Demmers, Y. Cao, S. Gauss, J. Lowe, D. Parsons, and C. Wathes. “Neural predictive control of broiler chicken growth”. *IFAC Proceedings Volumes*, 43(6):311–316, 2010. doi:10.3182/20100707-3-be-2012.0061.

K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer Science + Business Media, 2014. doi:10.1007/978-1-4471-5571-3.

J. Grahovac, A. Jokić, J. Dodić, D. Vučurović, and S. Dodić. “Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks”. *Renewable Energy*, 85:953–958, jan 2016. doi:10.1016/j.renene.2015.07.054.

A. e. F. N. o. Hasan Eleroğlu, Arda Yıldırım and M. Duman. “Comparison of growth curves by growth models in slow–growing chicken genotypes raised the organic system”. *International Journal of Agriculture and Biology*, 16(3):529–535, 2014.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.

S. C. Hernandez, J. A. Bueno, E. N. Sanchez, and L. Diaz-Jimenez. “State estimation by artificial neural networks in a continuous bioreactor”. *IFAC Proceedings Volumes*, 46(31):215–220, 2013. doi:10.3182/20131216-3-in-2044.00033.

A. Z. Lopes, L. Ferreira, T. Y. Junior, and W. S. Lacerda. “Modeling productive performance of broiler chickens with artificial neural network”. In *Livestock Environment VIII, 31 August - 4 September 2008, Iguassu Falls, Brazil*. American Society of Agricultural and Biological Engineers (ASABE), 2008. doi:10.13031/2013.25602.

V. V. Nair, H. Dhar, S. Kumar, A. K. Thalla, S. Mukherjee, and J. W. Wong. “Artificial neural network based modeling to evaluate methane yield from biogas in a laboratory-scale anaerobic bioreactor”. *Bioresource Technology*, 217:90–99, oct 2016. doi:10.1016/j.biortech.2016.03.046.

R. Nasimi and R. Irani. “Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks”. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14):1604–1611, may 2014. doi:10.1080/15567036.2011.592903.

D. Nguyen and B. Widrow. “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights”. In *1990 IJCNN International Joint Conference on Neural Networks*. Institute of Electrical & Electronics Engineers (IEEE), 1990. doi:10.1109/ijcnn.1990.137819.

OECD. “OECD-FAO agricultural outlook 2015”, jul 2015.

C. D. Rajamani Doraiswami and M. Stevenson. “Closed loop identification”. In *Identification of Physical Systems*, pages 357–378. Wiley-Blackwell, mar 2014. doi:10.1002/9781118536483.ch8.

K. M. Sameer Agarwal and Others. “Ceres solver”. `http://ceres-solver.org`, 2015.

K. Stacey, D. Parsons, A. Frost, C. Fisher, D. Filmer, and A. Fothergill. “An automatic growth and nutrition control system for broiler production”. *Biosystems Engineering*, 89(3):363–371, nov 2004. doi:10.1016/j.biosystemseng.2004.07.006.

H. Wang, X. Yan, H. Chen, C. Chen, and M. Guo. “Chlorophyll-a predicting model based on dynamic neural network”. *Applied Artificial Intelligence*, 29(10):962–978, nov 2015. doi:10.1080/08839514.2015.1097142.

C. Wathes, H. Kristensen, J.-M. Aerts, and D. Berckmans. “Is precision livestock farming an engineer’s daydream or nightmare, an animal’s friend or foe, and a farmer’s panacea or pitfall?”. *Computers and Electronics in Agriculture*, 64(1): 2–10, nov 2008. doi:10.1016/j.compag.2008.05.005.

Paper B

Broiler Weight Forecasting using Dynamic Neural Network Models with Input Variable Selection

Simon V. Johansen[a,b], Jan D. Bendtsen[b], Martin R.-Jensen[a] and Jesper Mogensen[b]

[a]SKOV A/S, Hedelund 4, Glyngøre, Denmark
[b]Department of Control and Automation, Aalborg University, Denmark

Abstract**—***The global demand for poultry meat is predicted to increase by 18% between 2015-17 and 2027, which motivates the need for better tools for production monitoring, planning and optimization. This paper presents the first results on broiler (chicken for meat production) weight forecasting intended for production planning and monitoring using environmental broiler house conditions – such as heating, ventilation, and temperature. We investigate the dynamic impact of environmental conditions on broiler growth, which is known to be highly significant but unexplored in scientific literature. The forecasting is carried out using ensemble dynamic neural network models trained on past production data with mutual information based input variable selection. To investigate the potential of the proposed method, an extensive case study on almost 3.5 years of industrial farm scale production data from a state-of-the-art broiler house is carried out. The dynamic impact of environmental conditions on broiler growth is found to be significant and useful broiler weight forecasts are obtained – effectively providing a foundation for future research on optimization of broiler production.*

Keywords**—***agricultural engineering; biological system modeling; recurrent neural networks*

Published in *Journal of Computers and Electronics in Agriculture*, 2019.

Contents

1 Introduction

Due to the growing middle class in many developing countries, the increase in meat consumption is predicted to continue rising [OECD, 2018, pp. 22]. The highest growth in meat production is foreseen in poultry meat, of which we expect broiler (chicken for meat production) meat to represent the majority. One reason being that broiler production has the highest yield per feed unit among land animals, making chicken meat a relatively inexpensive source of animal protein. The poultry industry on a world scale is predicted to steadily increase, from an average of 118.1 billion kg of meat in 2015-2017 to 139.0 billion kg in 2027 [OECD, 2018, pp. 37] – a significant predicted increase of 18%. This drives the need for better tools for production monitoring, planning and optimization, where this paper presents the first results on broiler weight forecasting intended for production planning and monitoring. The intended end goal is production optimization, where production planning tools will aid the development of production optimization strategies.

Empirically motivated nonlinear growth curve models have traditionally been used to determine the evolution of broiler weight. These models have been extensively studied in scientific literature such as [Aggrey, 2002], [Ahmadi and Mottaghitalab, 2007] and [Hasan Eleroğlu and Duman, 2014]. A growth curve in this context is a function of time with a fixed structure fitted to past broiler weight data. Common models include the Richerds Model and Gompertz-Laird Model that are described by 4 and 3 parameters respectively, where the parameters have biologically intuitive interpretations – such as time and size of maximum growth rate [Aggrey, 2002]. See [Demuner et al., 2017] for common growth curve models and parameters for the most popular broiler strains, including ROSS 308 used in this study. In [Lopes et al., 2008], the relationship between broiler house environment and production performance was investigated using a non-dynamic neural network. Likewise, in [Diez-Olivan et al., 2018] quantile regression forests-based modeling, a non-dynamic method, was among others used to predict broiler weight on week 3, 5 and 6 using broiler house environment. However, it is not clear how to extract temporal performance, i.e. day-to-day growth prediction, from such non-dynamic models.

Dynamic broiler growth models are an extension of growth curves and have primarily been developed for control synthesis in scientific literature. In [Aerts et al., 2003] a time-variant online parameter estimation of a dynamic model was successfully applied to predict future weight up to 7 days ahead without a priori information. The model uses feed intake as input and weight as output and was used for model predictive control (MPC) in [Cangar et al., 2007]. This control algorithm was tested in a laboratory setting with a stocking density of 5.3 birds/m^2 and 20 birds/m^2, the latter to emulate farm scale density. The mean relative weight control error was 2.7% and 7.3% for the low and high-density experiments respectively – suggesting that farm scale broiler production is harder to both predict and control. A similar result was obtained in

[Demmers et al., 2010], where a small differential recurrent neural network was used to model the feed quantity and control the broiler weight using nonlinear MPC. In [Stacey et al., 2004] a dynamic broiler weight model was developed and used information about feed uptake and composition of two feed types with known nutritional value. It was successfully used for broiler weight control and achieved results comparable to that of a stockman on farm scale broiler production with 30,000-40,000 broilers per house.

Hence, the available scientific literature on dynamic broiler models are very sparse and focuses exclusively on active broiler weight control by regulating feed uptake and composition. This is understandable, as feed is considered the biggest expense in broiler production. However, this research is incompatible with *state-of-the-art* broiler industry as it uses ad libitum feeding regimes, which excludes feed uptake regulation. Regulation of environmental broiler house conditions, such as temperature, are compatible with industry, and if managed correctly are known to minimize the required feed, water, and electricity to produce a mature broiler. No prior scientific modeling literature has been found studying the complex dynamic interconnection between broiler weight and broiler house environmental conditions, and will thus be the subject of this work. In this paper dynamic neural network (DNN) class models are used for forecasting and mutual information (MI) based input variable selection (IVS) for automatic input configuration.

State-of-the-art industrial broiler production typically executes 5-8 batches per house per year, of which identical broiler houses in terms of physical construction, management and climate nonetheless consistently exhibit different growth and feed uptake performance. Hence, data from multiple houses cannot easily be pooled together, which limits modeling to a single house. A parameter drift caused by naturally changing production conditions, such as the broiler house deteriorating and the broiler and feed performance increases, prevents indefinite data accumulation and renders old production data invalid. Hence, broiler production data is a very limited resource. Note that data quantity requirements increase exponentially with the number of inputs, input lags and output lags for the DNN [May et al., 2011]. To make best possible use of the limited production data, IVS is used to select as few significant inputs and lags as possible. No prior scientific literature on IVS of broiler growth and environmental variables has been found, and will also be a subject of this work.

IVS is a critical step in data driven modeling, where appropriate input variables are selected from available data. The IVS discipline is diverse and has many perspectives and applications – for an overview of time series IVS see [May et al., 2011], and for feature selection see [Brown et al., 2012]. Given our application, we narrow our scope to model free filter type IVS algorithms – also known as filter type IVS. One class of statistical IVS is based on MI. By definition, mutual information $I(X;Y)$ is the reduction in uncertainty with respect to the random variable Y due to the observation of the random variable X [May et al., 2008b]. MI is often calculated by estimating the underlying

marginal and joint probability density functions (PDFs) through kernel density estimation (KDE). Such an IVS algorithm was formulated in [Sharma, 2000], and modified in [May et al., 2008b] among others. It uses successive regression to remove selected information from the input candidates and target output and estimates partial mutual information (PMI), while relying on computationally heavy bootstrapping to determine a stopping criteria. Alternatively, the Copula entropy can be used to estimate MI[Chen et al., 2014].

Dynamic neural networks have been successfully applied to model complex biological processes. Recent applications include algae growth prediction in a laboratory setting [Wang et al., 2015], prediction of bioethanol production in a bioreactor [Grahovac et al., 2016], bioreactor prediction [Nair et al., 2016], yeast fermentation modeling in a bioreactor [Nasimi and Irani, 2014], state estimation in a continuous bio reactor [Hernandez et al., 2013], and prediction of dissolved oxygen in wastewater treatment [Sadeghassadi et al., 2018]. Likewise, mutual information based IVS has been applied extensively to environmental modeling, such as IVS for prediction of rainfall [Sharma, 2000], salinity [Bowden et al., 2005], Water Quality [May et al., 2008a], storm water runoff [He et al., 2011], flood forecasting [Chen et al., 2014], and rainfall-runoff [Li et al., 2015]. Other recent applications include return temperature estimation in mixing loops [Overgaard et al., 2017], and Coriolis flow-meters for two-phase flow [Wang et al., 2017].

In the present effort, we present a data-driven dynamic neural network broiler weight forecasting model with MI based IVS. In particular, we show that dynamic interactions between environmental conditions and broiler weight in ad libitum feed broilers are present and can be captured by such a model. We do this in a data driven framework on real farm scale production data from a state-of-the-art broiler house. Preliminary results were presented in [Johansen et al., 2017]. Hence, the scientific contribution of this paper is a new application of dynamic neural networks and MI based IVS for broiler weight forecasting, in addition to an IVS analysis of broiler weight and environmental variables.

The remainder of the paper is structured as follows. The DNN model is described in Section 2, the MI based IVS is described in Section 3, and the DNN model and IVS are combined to a forecasting algorithm in Section 4 along with the forecasting evaluation criteria. The experimental verification setup is described in Section 5, consisting of a single forecasting trial in Section 6, and an analysis of all trials in Section 7. Concluding remarks are provided in Section 8.

2 Dynamic Neural Network Model (DNN)

2.1 Model

Since the multilayer perceptron (MLP) model and its variations have been extensively researched in scientific literature we will only give a brief introduction

and describe how we apply it to our specific problem. For a thorough introduction to the subject we redirect the reader to [Du and Swamy, 2014] and [Haykin, 1994].

The particular type of Dynamic Neural Network (DNN) model used in this work can be classified as a discrete-time nonlinear ARMAX model given by

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = \mathcal{N}\big(\hat{Y}[k], U[k] \mid \mathcal{W}\big) \qquad \forall p \le k \tag{1a}$$

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = y[k+1 \mid b] \qquad \forall k < p \tag{1b}$$

with

$$U[k] = \begin{bmatrix} u[k-\bar{n}_1 \mid b]^T & \cdots & u[k-\bar{n}_{N_{\bar{n}}} \mid b]^T \end{bmatrix}^T$$

$$\hat{Y}[k] = \begin{bmatrix} \hat{y}[k-\bar{m}_1 \mid b,p,\mathcal{W}]^T & \cdots & \hat{y}[k-\bar{m}_{N_{\bar{m}}} \mid b,p,\mathcal{W}]^T \end{bmatrix}^T$$

where $\hat{y}[k \mid b,p,\mathcal{W}] \in \mathbb{R}^{N_y}$ is the model output at sample $k \in \mathbb{Z}_+$, initialized with data from batch $b \in \mathbb{Z}$ at sample $p \in \mathbb{Z}_+$ with the $N_{\mathcal{W}}$ model weights $\mathcal{W} \in \mathbb{R}^{N_{\mathcal{W}}}$, $y[k \mid b] \in \mathbb{R}^{N_y}$ is the measured output and $u[k \mid b] \in \mathbb{R}^{N_u}$ is the measured input at batch b and sample k, $\mathcal{N}\colon \mathbb{R}^{N_y N_{\bar{m}}} \times \mathbb{R}^{N_u N_{\bar{n}}} \to \mathbb{R}^{N_y}$ is a MLP model, $U[k] \in \mathbb{R}^{N_u N_{\bar{n}}}$ is delayed values of the input vector $u[k \mid b]$ corresponding to the $N_{\bar{n}} \in \mathbb{Z}_+$ elements of $\bar{n} = \{\bar{n}_1, \ldots, \bar{n}_{N_{\bar{n}}}\}$, and $\hat{Y}[k] \in \mathbb{R}^{N_y N_{\bar{m}}}$ is delayed values of the past output vector $\hat{y}[k \mid b,p,\mathcal{W}]$ corresponding to the $N_{\bar{m}} \in \mathbb{Z}_+$ elements of $\bar{m} = \{\bar{m}_1, \ldots, \bar{m}_{N_{\bar{m}}}\}$. The sample number k and initialization sample p for the model output $\hat{y}[k \mid b,p,\mathcal{W}]$ is bounded by $p \in [1; N_{s,b}]$ and $k \in [p+1; N_{s,b}]$ where $N_{s,b} \in \mathbb{Z}_+$ is the number of samples in batch b. This particular structure has been adopted to accommodate potentially long propagation delays, while keeping the number of weights relatively low. Model initialization occurs through (1b), where k is implicitly lower bounded by 1 for both $y[k \mid b]$ and $u[k \mid b]$.

The DNN model $\mathcal{N}$ is selected with one hidden layer with hyperbolic tangent activation function in the hidden layer and linear activation function in the output layer. It can be shown that $\mathcal{N}$ is an universal function approximator, meaning that it has the capacity of approximating any system at any accuracy [Du and Swamy, 2014, chap. 4.2]. In matrix-vector representation, (1a) is written explicitly as

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = W^{\mathrm{o}} \tanh\big(\mathcal{X} + \theta^{\mathrm{h}}\big) + \theta^{\mathrm{o}} \tag{3}$$

with

$$\mathcal{X} = \sum_{i=1}^{N_{\bar{m}}} W^{\mathrm{h}}_{y,i} \hat{y}[k-\bar{m}_i \mid b,p,\mathcal{W}] + \sum_{j=1}^{N_{\bar{n}}} W^{\mathrm{h}}_{u,j} u[k-\bar{n}_j \mid b],$$

where $N_h \in \mathbb{Z}_+$ is the number of neurons in the hidden layer, $\mathcal{X} \in \mathbb{R}^{N_h}$, $W^{\mathrm{o}} \in \mathbb{R}^{N_y \times N_h}$ is the output weights, $W^{\mathrm{h}}_{y,i} \in \mathbb{R}^{N_h \times N_y}$ is the delayed output weights, $W^{\mathrm{h}}_{u,j} \in \mathbb{R}^{N_h \times N_u}$ is the delayed input weights, $\theta^{\mathrm{h}} \in \mathbb{R}^{N_h}$ is the hidden layer bias

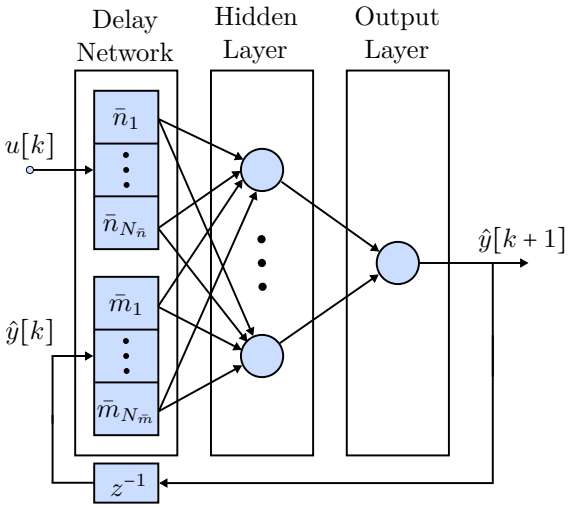


Figure 1: Visual representation of (3) with one input and one output for simplicity. Neurons are represented with blue circles, which contain an activation function and bias each. Note that all incoming signals to a neuron are multiplied by a weight. The operator z^{-1} produces a delay of 1 sample, while a blue box produce a delay equal to its number.

and $\theta^{\text{o}} \in \mathbb{R}^{N_y}$ is the output bias. A visual representation of (3) is depicted on Figure 1.

Input and output specific delays selected by the IVS are accommodated by modifying the structure of $W^{\text{h}}_{u,j}$ and $W^{\text{h}}_{y,i}$. For example, if the inputs indexed 1 and 3 are selected with delay of $j = 2$, $N_u = 4$ inputs, $N_h = 3$ hidden neurons, then $W^{\text{h}}_{u,j}$ equals

$$W^{\text{h}}_{u,2} = \begin{bmatrix} \mathcal{W}_1 & 0 & \mathcal{W}_2 & 0 \\ \mathcal{W}_3 & 0 & \mathcal{W}_4 & 0 \\ \mathcal{W}_5 & 0 & \mathcal{W}_6 & 0 \end{bmatrix}. \tag{4}$$

2.2 Model Training

The dynamic neural network is trained by iteratively minimizing the cost function

$$E_{\mathcal{B}}(\mathcal{W}) = \overbrace{\sum_{b\in\mathcal{B}} \sum_{p=1}^{N_P} \sum_{k=P_p}^{N_{s,b}} \frac{\|y[k \mid b] - \hat{y}[k \mid P_p, b, \mathcal{W}]\|_2^2}{N_{r,\mathcal{B}}}}^{E_{r,\mathcal{B}}} + \bar{\alpha} \overbrace{\|\mathcal{W}\|_2^2}^{E_{\mathcal{W}}} \tag{5}$$

with

$$N_{r,\mathcal{B}} = \sum_{b\in\mathcal{B}} \sum_{p=1}^{N_P} \sum_{k=P_p}^{N_{s,b}} N_y = N_{\mathcal{B}} N_P N_y (N_{s,b} + 1) - N_{\mathcal{B}} N_y \sum_{p=1}^{N_P} P_p,$$

where $\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_{N_\mathcal{B}}\}$ denotes the $N_\mathcal{B} \in \mathbb{Z}_+$ training batch indexes used for training, $N_{s,b} \in \mathbb{Z}_+$ is the number of samples in batch b, and $N_{r,\mathcal{B}} \in \mathbb{Z}_+$ is the total number of residuals in $E_{r,\mathcal{B}} \in \mathbb{R}_+$ with batch indexes $\mathcal{B}$. $\|\cdot\|_2$ is the standard Euclidean 2-norm. Both the inputs and outputs are normalized to a mean of 0 and standard deviation of 1 during training. The network weights $\mathcal{W}$ are initialized using the Nguyen-Widrow algorithm as explained in [Nguyen and Widrow, 1990].

Each batch is divided into $N_P \in \mathbb{Z}_+$ Sub-Batches with initial starting times denoted by $P = \{P_1, \ldots, P_{N_P}\}$, which we find speeds up training and decreases the risk of converging to a poor local minima. Consequently, for each batch $b \in \mathcal{B}$ we get N_P sets of trajectories with different initial conditions to minimize, as illustrated on Figure 2, where each output is weighted equally.

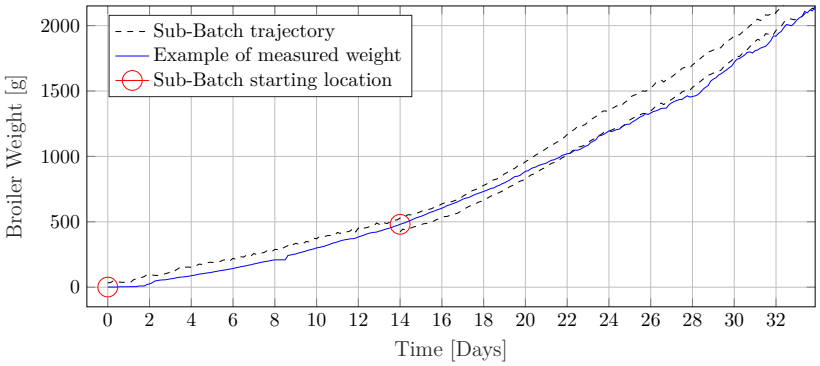


Figure 2: Example of the Sub-Batch trajectories with Sub-Batch starting time at day $P = \{0, 14\}$.

The last term of the cost function is a scalar regularization term punishing the size of the $N_\mathcal{W}$ system weights $\mathcal{W}$, where $\bar{\alpha} \in \mathbb{R}_+$ is the regularization weight. The regularization weight $\bar{\alpha}$ is determined iteratively through Bayesian Regulation as described in [Burden and Winkler, 2008]. We normalize the regularization weight to make the cost functions comparable between trainings, and is calculated according to $\bar{\alpha} = \alpha/\beta$ with:

$$\alpha = \frac{\gamma}{2E_\mathcal{W}} \qquad \beta = \frac{N_{r,\mathcal{B}} - \gamma}{2E_{r,\mathcal{B}}} \qquad \gamma = N_\mathcal{W} - \alpha \operatorname{trace}\left(G^{-1}\right) \tag{6}$$

Where $E_\mathcal{W} \in \mathbb{R}_+$ and $E_{r,\mathcal{B}}$ originate from (5) and γ, $0 \leq \gamma \leq N_\mathcal{W}$, is a measure of how many of the $N_\mathcal{W}$ parameters are unused. Lastly, $G \in \mathbb{R}^{N_\mathcal{W} \times N_\mathcal{W}}$ is the Hessian matrix of the joint cost $\beta E_{r,\mathcal{B}} + \alpha E_\mathcal{W}$ as calculated in (7a), where I is the identity matrix of appropriate dimensions. It is partially approximated through the Jacobian $J_{r,\mathcal{B}} \in \mathbb{R}^{N_{r,\mathcal{B}} + N_\mathcal{W} \times N_\mathcal{W}}$ of $E_{r,\mathcal{B}}$ in (7b), similar to the Levenberg Marquard algorithm. The scalar constants $\alpha \in \mathbb{R}_+$, $\beta \in \mathbb{R}_+$ and $\bar{\alpha} \in \mathbb{R}_+$ are iteratively updated after each training iteration, with initial conditions $\alpha = 0$

and $\beta = 1$.

$$G = \frac{d^2(\beta E_{r,\mathcal{B}} + \alpha E_{\mathcal{W}})}{d\mathcal{W}\, d\mathcal{W}^T} = \frac{d^2(\beta E_{r,\mathcal{B}})}{d\mathcal{W}\, d\mathcal{W}^T} + 2\alpha I \tag{7a}$$

$$\approx J_{r,\mathcal{B}}^T J_{r,\mathcal{B}} + 2\alpha I \text{ with } J_{r,\mathcal{B}} = \beta \frac{dE_{r,\mathcal{B}}}{d\mathcal{W}^T} \tag{7b}$$

Cross validation in combination with early stopping is applied to avoid overfitting and increase the models ability to generalize to inputs not present in the training data. This is facilitated by selecting the model weights $\mathcal{W}$ among all training iterations with the smallest cost $E_{\tilde{b}}(\mathcal{W})$ calculated with the testing batch $\tilde{b} \in \mathbb{Z}$. Note that the testing batch is only used for early stopping and not for training of $\mathcal{W}$, i.e. $\tilde{b} \notin \mathcal{B}$. This type of cross validation could accurately be termed leave-one-batch-out cross validation, but is a special case of the holdout cross validation method – see [Arlot and Celisse, 2010] for a comprehensive review of cross validation methods.

The cost function (5) is minimized using the Levenberg-Marquardt (LM) optimization algorithm by means of the Ceres Solver library [Sameer Agarwal and Others, 2015]. Ceres implements an exact step LM variant based on [Madsen et al., 2004]. It is an algorithm that efficiently solves large scale least square problems on the form

$$\mathcal{W}(\mathcal{B}, \tilde{b}) = \arg\min_{\mathcal{W}} \left[\frac{1}{2} E_{\mathcal{B}}(\mathcal{W}) = \|F_{\mathcal{B}}(\mathcal{W})\|_2^2 \right], \tag{8}$$

where $E_{\mathcal{B}}: \mathbb{R}^{N_{\mathcal{W}}} \to \mathbb{R}$ is the cost function to be minimized, $F_{\mathcal{B}}: \mathbb{R}^{N_{\mathcal{W}}} \to \mathbb{R}^{N_{r,\mathcal{B}}+N_{\mathcal{W}}}$ is a vector with the $N_{r,\mathcal{B}}+N_{\mathcal{W}}$ residuals of $E_{\mathcal{B}}$, $\mathcal{W} \in \mathbb{R}^{N_{\mathcal{W}}}$ is the weights and $\mathcal{W}(\mathcal{B}, \tilde{b}) \in \mathbb{R}^{N_{\mathcal{W}}}$ is the optimal weights found by the algorithm with training batches $\mathcal{B}$ and early stopping using batch $\tilde{b}$ as discussed previously. Note that the problem is not guaranteed to be convex, and consequently $\mathcal{W}(\mathcal{B}, \tilde{b})$ is only guaranteed to be a local minimum. In the following the dependence on $\mathcal{W}$ is implicit for the residuals $F_{\mathcal{B}}(\mathcal{W})$ and its Jacobian $J_{F_{\mathcal{B}}}: \mathbb{R}^{N_{\mathcal{W}}} \to \mathbb{R}^{(N_{r,\mathcal{B}}+N_{\mathcal{W}}) \times N_{\mathcal{W}}}$ given by $J_{F_{\mathcal{B}}}(\mathcal{W}) = \frac{dF_{\mathcal{B}}(\mathcal{W})}{d\mathcal{W}^T}$. LM is an iterative procedure and requires an initial weight guess. The weights are updated according to $\mathcal{W} \leftarrow \mathcal{W} + \Delta\mathcal{W}$ by solving for the weight change $\Delta\mathcal{W} \in \mathbb{R}^{N_{\mathcal{W}}}$ in

$$\left(J_{F_{\mathcal{B}}}^T J_{F_{\mathcal{B}}} + \mu I\right) \Delta\mathcal{W} = -J_{F_{\mathcal{B}}}^T F_{\mathcal{B}}, \tag{9}$$

where $\mu \in \mathbb{R}_+$ is the damping parameter, using the Cholesky Factorization. The LM algorithm is a trust region method and controls the step size with the damping parameter μ, of which a larger value will reduce the step size. This is controlled by the ratio between the actual and predicted cost change given by

$$\rho = \frac{E_{\mathcal{B}}(\mathcal{W}) - E_{\mathcal{B}}(\mathcal{W} + \Delta\mathcal{W})}{\hat{E}_{\mathcal{B}}(0) - \hat{E}_{\mathcal{B}}(\Delta\mathcal{W})}, \tag{10}$$

where 0 is the zero-vector, and $\hat{E}_{\mathcal{B}}(\Delta\mathcal{W}) \approx E_{\mathcal{B}}(\mathcal{W}+\Delta\mathcal{W})$ is an approximation of $E_{\mathcal{B}}(\mathcal{W}+\Delta\mathcal{W})$ in $\mathcal{W}$ given by

$$\hat{E}_{\mathcal{B}}(\Delta\mathcal{W}) = E_{\mathcal{B}}(\mathcal{W}) + \Delta\mathcal{W}^T J_{F_{\mathcal{B}}}^T F_{\mathcal{B}} + \Delta\mathcal{W}^T J_{F_{\mathcal{B}}}^T J_{F_{\mathcal{B}}} \Delta\mathcal{W}.$$

If the cost function is successfully decreased by $\Delta\mathcal{W}$, then the step is accepted and the damping parameter μ is updated according to

$$\mu \leftarrow \mu \cdot \max\left\{\tfrac{1}{3}, 1-(2\rho-1)^3\right\}, \tag{11}$$

otherwise the step is rejected and the damping parameter is increased according to $\mu \leftarrow \mu \cdot 2^\eta$ with $\eta \in \mathbb{Z}_+$ being the number of consecutive rejected updates. Note that if $\frac{1}{2} < \rho$ then μ is decreased, which increases the trust region, as it indicates that $J_{F_{\mathcal{B}}}$ is a good approximation. This process is repeated until $\mathcal{W}$ converges, or 1000 training iterations have elapsed.

3 Mutual Information-based Input Selection

3.1 Information Theory

The information entropy of the random variable X is given by

$$H(X) = -\int_X p(x)\ln p(x)dx, \tag{12}$$

and is a measure of how much information is contained by X. Note that this measure extends to multivariate inputs as well, in which case it is a measure of the collective information of all variables. The mutual information $I(X;Y)$ is the reduction in uncertainty with respect to the random variable Y due to the observation of the random variable X, and is given by

$$I(X;Y) = -\int_Y \int_X p(x,y)\ln\frac{p(x,y)}{p(x)p(y)}dx\,dy. \tag{13}$$

Alternatively, it can be regarded as the amount of information X provides about Y, and vice versa. It can be shown to be a function of the linear correlation coefficient $R_{XY} \in [-1;1]$ in case of linear and Gaussian distributed inputs given by

$$I(X;Y) = -\frac{1}{2}\ln\left(1-R_{XY}^2\right). \tag{14}$$

However, mutual information is more robust due to its insensitivity to noise and data transformations, i.e. nonlinearity [May et al., 2011]. Through resubstitution (12) and (14) can be estimated by

$$\hat{H}(X) = -\frac{1}{n}\sum_{k=1}^{n}\ln\hat{f}_X(X[k]) \text{ and} \tag{15a}$$

$$\hat{I}(X;Y) = \frac{1}{n}\sum_{k=1}^{n}\ln\frac{\hat{f}_{X,Y}(X[k],Y[k])}{\hat{f}_X(X[k])\hat{f}_Y(Y[k])}, \tag{15b}$$

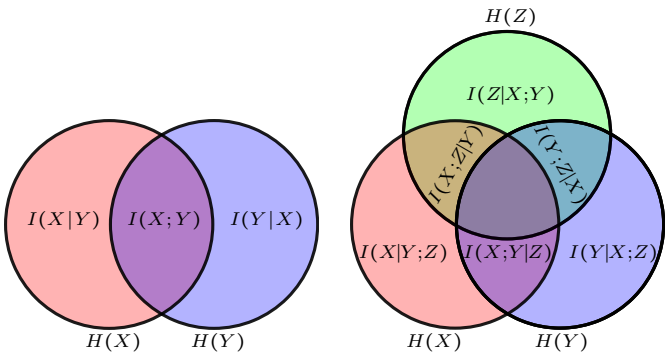
where $\hat{f}_X(\cdot)$ and $\hat{f}_Y(\cdot)$ are the estimated marginal density function of X and Y, respectively, and $\hat{f}_{X,Y}(\cdot,\cdot)$ is the estimated joint density function of X and Y [Beirlant et al., 1997]. Intuitively, the nominator of (15b) equals the denominator if X and Y are uncorrelated, as $\hat{f}_X(X[k])\hat{f}_Y(Y[k]) = \hat{f}_{X,Y}(X[k], Y[k])$, and consequently does not add any new information, as $\ln 1 = 0$. With $X = \{X_1, \ldots, X_i\}$, $Y = \{Y_1, \ldots, Y_j\}$ and $x \in \mathbb{R}^i$ the notation for multivariate mutual information and i-variate joint density estimation is

$$I(X;Y) \triangleq I(X_1, \ldots, X_i; Y_1, \ldots, Y_j) \text{ and}$$
$$\hat{f}_X(x) \triangleq \hat{f}_{X_1,\ldots,X_i}(x_1, \ldots, x_i).$$

The partial mutual information (PMI) is a measure of the information between two variables X and Y, excluding the information from a third variable Z. It can be regarded as a measure of how much mutual information is not explained by Z, which is useful for determining if an input variable provides additional information about an output variable. In terms of mutual information and information entropy it is given by:

$$\begin{aligned} I(X;Y \mid Z) &= I(X;Y,Z) - I(X;Z) \\ &= H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) \end{aligned} \tag{16}$$

The relationship between information entropy, mutual information and partial mutual information is visualized on Figure 3. Applying (19) on a dataset is illustrated on Figure 4.



(a) MI **(b)** PMI

Figure 3: Venn diagram representation of MI and PMI.

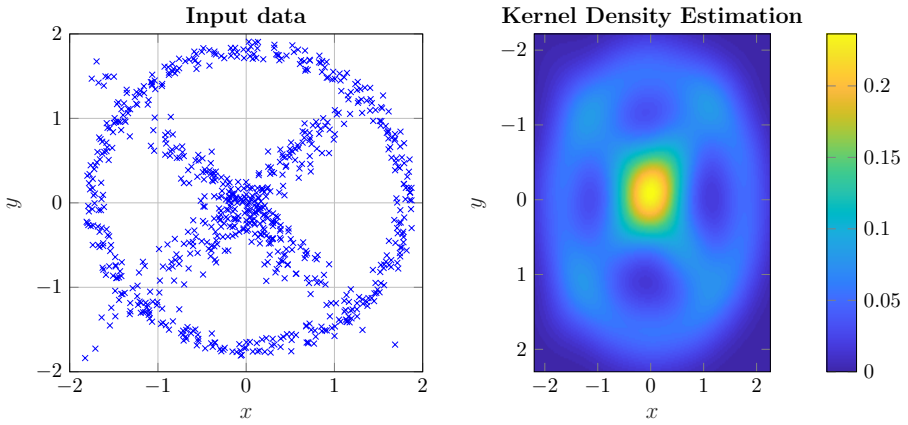


Figure 4: The estimated probability density distribution $\hat{f}_{X,Y}(x, y)$ given by (19) of two correlated variables x and y. The linear correlation coefficient is -0.005 in this case, suggesting that x and y are very close to being uncorrelated. The KDE represents the underlying correlated distribution significantly better in this case. The Mutual Information $\hat{I}(x; y) = 0.204$ given by (15b), suggesting that information of y can be obtained from x, as the 95% critical value for Gaussian noise with $n = 400$ samples equals 0.0567 [May et al., 2008a].

3.2 Kernel Density Estimation (KDE)

To calculate the joint PDFs in (15), we employ kernel density estimation. The d-variate Gaussian kernel density estimator (KDE) can be formulated by

$$\hat{f}_X(x) = \frac{1}{n}\sum_{k=1}^{n} K_{\bar{H}}(x - X[k]) \text{ with} \tag{17a}$$

$$K_{\bar{H}}(y) = \frac{K\left(\bar{H}^{-1}y\right)}{\det\left\{\bar{H}\right\}} \text{ and} \tag{17b}$$

$$K(z) = \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2}z^T z\right). \tag{17c}$$

Where (17a) and (17b) are the multivariate parzen window with $x, y, z \in \mathbb{R}^d$, $\bar{H} \in \mathbb{R}^{d\times d}$ being the kernel bandwidth or smoothing matrix, n is the number of samples in the known data $X \in \mathbb{R}^{d\times n}$, and $X[k] \in \mathbb{R}^d$ denotes the k'th sample of X. In this work we use the generalized Scott's rule [Wolfgang Härdle and Werwatz, 2004, pp. 73] given by

$$\bar{H} = \Sigma^{1/2}\,\bar{h}_S \text{ with } \bar{h}_S = n^{-1/(d+4)}, \tag{18}$$

where $\Sigma \in \mathbb{R}^{d\times d}$ is the covariance matrix of the known data $X \in \mathbb{R}^{d\times n}$ and $\bar{h}_S \in \mathbb{R}$. Furthermore, (17c) is the kernel function, where we use the multivariate

Gaussian density function. The full expression for (17) equals

$$\hat{f}_X(x) = \sum_{k=1}^{n} \frac{\exp\left(-\frac{1}{2}(x - X[k])^T \left(\bar{h}_S^2 \Sigma\right)^{-1} (x - X[k])\right)}{n\sqrt{(2\pi)^d \det\left(\bar{h}_S^2 \Sigma\right)}}. \tag{19}$$

[Li et al., 2015] investigates different kernel bandwidth selection algorithms for the IVS algorithm in [May et al., 2008b] under non-normal conditions, as [May et al., 2008b] assumes normality through the Gaussian reference rule. Using alternative bandwidth selection algorithms lead to increased selection accuracy in IVS for rainfall and runoff prediction. However, we have not found it necessary to investigate alternative bandwidth selection algorithms for our purposes – suggesting that the data might be close to Gaussian.

3.3 Input Variable Selection (IVS)

The task of progressively selecting input variables can be formulated as follows. Let X denote the set of lagged input candidates, $S = \{S_1, \ldots, S_{m-1}\}$ the $(m-1)$ previously selected lagged inputs with $S \subseteq X$ and Y being the target output. The m'th lagged input candidate with *most new information* is selected according to

$$S_m = \underset{C \in X \setminus S}{\arg\max}\, I(C, S; Y) = \underset{C \in X \setminus S}{\arg\max}\, I(C; Y \mid S). \tag{20}$$

This is repeated until S_m provides *no significant* new information about Y, and S_m is not added to S. Originally, the 95 percentile critical bootstrap value was used to estimate a termination threshold, but since it is computationally heavy [May et al., 2008b] compared alternatives. Termination criterias based on the Akaike Information Criterion, modified Hampel outlier test and PMI lookup table based on Monte Carlo simulations of normally distributed noise for different sample sizes were compared. We have good experience with using the 95-percentile tabulated PMI as depicted on Figure 5, but it should be noted that [May et al., 2008b] found that this method gives biased values for non-Gaussian distributed inputs. Realizing that this table is approximately log-log linear in the number of samples n, the following approximation is found (See Figure 5):

$$\hat{I}^{(95)}(n) = 1.7134 \cdot n^{-0.518} \tag{21}$$

Calculation of (20) requires estimation of an $(m+1)$ dimensional probability density function using (19), and is therefore increasingly dependent on the data quality for larger m – both in terms of number of samples and increasing data sparsity in higher dimensional spaces. This phenomenon is called *the curse of dimensionality* and is illustrated in Table 1. Hence, calculation of higher order MI is not preferred, as we cannot make any guarantees regarding data quality for broiler production.

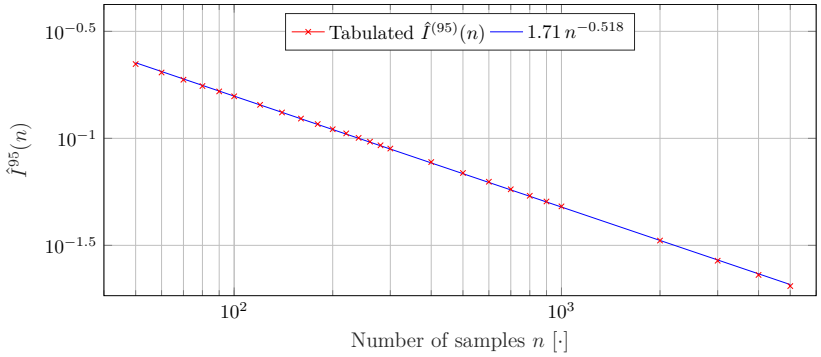


Figure 5: Tabulated PMI and an approximation [May et al., 2008b].

Dimension	Sample Size
1	4
2	19
3	67
4	223
5	768
6	2790
7	10700
8	43700
9	180700
10	842000

Table 1: Sample size required to maintain a constant standard error in a general regression neural network [May et al., 2011].

3.4 Regression based PMI

This method is taken from [May et al., 2008b] and uses the KDE based conditional expectation known as the Nadaraya-Watson estimator for regression. It is the regression of Y given the observed X, denoted by $E[Y\,|\,X]$, where the j'th sample is calculated by

$$E[Y \mid X = X[j]] = \frac{1}{n} \frac{\sum_{k=1}^{n} Y[k] K_{\bar{H}}(X[j] - X[k])}{\sum_{k=1}^{n} K_{\bar{H}}(X[j] - X[k])}. \tag{22}$$

For simplicity, the removal of known "information" through regression of the elements of B from the variable A is written as

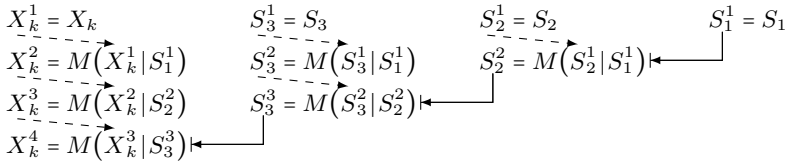
$$M(A|B) = A - E[A|B].$$

The key idea behind this method is to estimate the partial mutual information by removing the known mutual information from the selected inputs

through successive regression. Let $(\cdot)^i$ denote the i'th successive regression of $(\cdot)$, then the m'th iteration of the selected inputs $S = \{S_1^1, \ldots, S_{m-1}^{m-1}\}$ of $X_k \in X$ equals

$$X_k^m = \begin{cases} M\left(X_k^{m-1} | S_{m-1}^{m-1}\right) & m > 1 \\ X_k & m = 1 \end{cases}. \tag{23}$$

Notice that X_k^m depends on up to m regressions – one for each previously selected element of S. To demonstrate, X_k^4 is obtained from the following successive regressions of the previously selected variables, $\{S_3, S_2, S_1\}$, according to:



Note that the first row contains the original values, and each subsequent row is a regression based only on the previous row.

Using (23), the IVS problem formulated in (20) can be approximated by

$$S_m = \underset{C \in X \setminus S}{\arg\max} \left[I(C; Y \mid S) \approx I\left(C^{m-1}; Y^{m-1}\right) \right], \tag{24}$$

which has the benefit of only requiring 2-dimensional density function estimates compared to the previous $(m-1)$. When implementing this algorithm, the values of $(\cdot)^m$ are calculated from $(\cdot)^{m-1}$ after S_m has been selected to avoid unnecessary recalculation.

4 Broiler Weight Forecasting and Evaluation

This section describes how the DNN model in Section 2 and the MI based IVS in Section 3 are used for broiler weight forecasting along with how the forecasts are evaluated.

4.1 Forecasting Trial Setup

The aim of forecasting trial $t \in \{1, \ldots, N_t\}$ is to forecast future outputs $p{<}k{<}N_{s,t}$ from the present moment $p \in \mathbb{Z}_+$ throughout trial t, only based on past batches $\{1{-}N_{PB}, \ldots, t{-}1\}$ with N_{BP} preliminary batches prior to trial $t{=}1$. The batches are segmented into the following categories, as depicted on Figure 6, and equals: evaluation batch $\{t\}$, prediction batch $\{t{-}1\}$, testing batches $\{t{-}2\}$, and the remaining batches are denoted training batches $\{1{-}N_{PB}, \ldots, t{-}3\}$. The training and testing batches are used to find the network weights $\mathcal{W}$ through training. The latest training batch is used in an early stopping setting to avoid

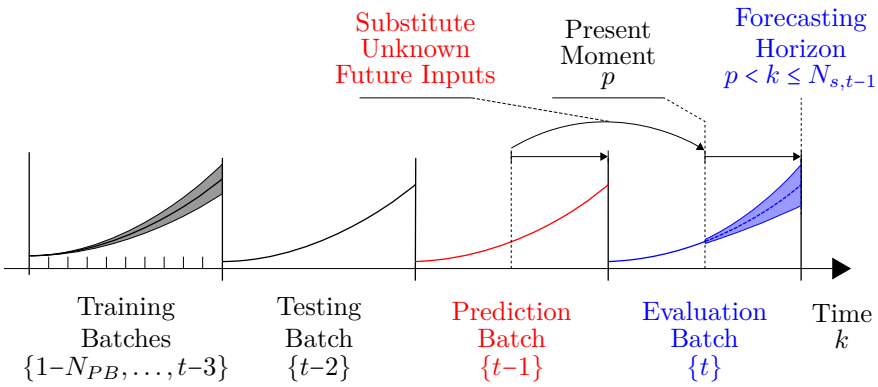


Figure 6: Visual representation of how batches are selected for forecasting trial t. The training and testing batches $\{N_{PB},\ldots,t-2\}$ are used to train a model as described in Section 2. When forecasting future outputs at the evaluation batch $\{t\}$ ahead of the present moment $p < k$, then substitute future inputs from the prediction batch $\{t-1\}$ are used up to sample $N_{s,t-1}$ in the evaluation batch $\{t\}$.

over-fitting. In order to forecast future outputs on the evaluation batch $\{t\}$, unknown future input values are required of which we use the inputs from the prediction batch $\{t-1\}$ as substitute up to sample $k \leq N_{s,t-1}$.

4.2 DNN Model IVS

IVS for the DNN model described in Section 2 requires selection of lags from both inputs and outputs, due to its recurrent structure. As all inputs and outputs are not guaranteed to be present in all the batches for trial t, up to $N_B \in \mathbb{Z}_+$ potential batches are selected for the IVS algorithm by maximizing

$$\begin{aligned} \mathcal{B}_t = \arg\max_{\tilde{\mathcal{B}}} \; & N_{\tilde{u}}(\tilde{\mathcal{B}})\, N_{\tilde{y}}(\tilde{\mathcal{B}})\, \min\{N_{\tilde{B}}(\tilde{B}), N_B\} + \frac{1}{\max\{\tilde{\mathcal{B}}\} - \min\{\tilde{\mathcal{B}}\} + 1} \\ \text{s.t.} \quad & \tilde{\mathcal{B}} \subseteq \{1 - N_{PB},\ldots,t-3\} \end{aligned} \tag{25}$$

where $\mathcal{B}_t$ is the set of batches used for IVS and training of trial t, $\tilde{\mathcal{B}}$ is a set of potential batch indexes, $N_{\tilde{u}}(\tilde{\mathcal{B}})$ and $N_{\tilde{y}}(\tilde{\mathcal{B}})$ are the number of potential inputs and outputs with batch indexes $\tilde{\mathcal{B}}$. As the broiler house naturally changes over time, a forgetting factor is introduced by upper bounding the number of potential training batches to N_B through the term $\min\{N_{\tilde{B}}(\tilde{B}), N_B\}$. The term $\frac{1}{\max\{\tilde{\mathcal{B}}\} - \min\{\tilde{\mathcal{B}}\} + 1} < 1$ ensures that the most recent batches take priority if more than N_B potential batches can be selected, where $\min\{\tilde{\mathcal{B}}\}$ is the smallest and $\max\{\tilde{\mathcal{B}}\}$ is the biggest index in $\tilde{\mathcal{B}}$.

The input candidates X are constructed from both the potential input

$\tilde{u} \in \mathbb{R}^{N_{\tilde{u}}(\mathcal{B}_t)}$ and output $\tilde{y} \in \mathbb{R}^{N_{\tilde{y}}(\mathcal{B}_t)}$ candidates from $\mathcal{B}_t$ according to

$$X = \left\{\tilde{u}_i\left[k-\tilde{k}\right] \mid \tilde{k} \in [1; N_{L,\tilde{u}}] \wedge i \in [1; N_{\tilde{u}}(\mathcal{B}_t)]\right\} \cup \cdots \\ \left\{\tilde{y}_i\left[k-\tilde{k}\right] \mid \tilde{k} \in [1; N_{L,\tilde{y}}] \wedge i \in [1; N_{\tilde{y}}(\mathcal{B}_t)]\right\} \tag{26}$$

where $\tilde{u}_i[k-\tilde{k}]$ or $\tilde{y}_i[k-\tilde{k}]$ denotes the i'th element of $\tilde{u}$ or $\tilde{y}$ delayed by $\tilde{k} \in \mathbb{Z}_+$ samples, and $N_{L,\tilde{y}} \in \mathbb{Z}_+$ and $N_{L,\tilde{u}} \in \mathbb{Z}_+$ denotes the maximum output and input lag, respectively. This results in a total of $N_{L,\tilde{u}}N_{\tilde{u}}(\mathcal{B}_t)+N_{L,\tilde{y}}N_{\tilde{y}}(\mathcal{B}_t)$ candidates. When calculating the KDE from X using (19), data from all batches are used simultaneously. To account for the largest lag, equivalent of $N_{L,\max} = \max\{N_{L,\tilde{u}}, N_{L,\tilde{y}}\}$, the first $N_{L,\max}$ samples are excluded from each batch – resulting in a total of $\sum_{b\in\mathcal{B}_t}(N_{s,b} - N_{L,\max})$ samples.

A separate IVS for each $\tilde{y}_i \in \tilde{y}$ is carried out, where IVS with $Y = \tilde{y}_i$ of the potential candidates X from (26) is denoted ${}^iS$. The overall IVS results in

$$\mathcal{S}_t = \bigcup_{i=1}^{N_{\tilde{y}}(\mathcal{B}_t)} {}^iS, \tag{27}$$

which is used to configure the DNN model given by (1a) as follows. The resulting selected outputs and inputs used by the DNN model are given by

$$u = \{\bar{u} \in \tilde{u} \mid k \in [0; N_{L,\tilde{u}}] \wedge \bar{u}[n-k] \in \mathcal{S}_t\} \text{ and} \tag{28a}$$

$$y = \tilde{y}. \tag{28b}$$

The input and output lags in (3) are given by

$$\bar{n} = \left\{\tilde{k} \in [0; N_{L,\tilde{u}}] \mid i \in [1; N_{\tilde{u}}(\mathcal{B}_t)] \wedge u_i[k-\tilde{k}] \in \mathcal{S}_t\right\} \tag{29a}$$

$$\bar{m} = \left\{\tilde{k} \in [0; N_{L,\tilde{y}}] \mid i \in [1; N_{\tilde{y}}(\mathcal{B}_t)] \wedge y_i[k-\tilde{k}] \in \mathcal{S}_t\right\} \tag{29b}$$

The weights $W^{\mathrm{h}}_{u,\tilde{k}}$ and $W^{\mathrm{h}}_{y,\tilde{k}}$ from (3) is configured as follows. Column $i \in [1; N_u]$ of $W^{\mathrm{h}}_{u,\tilde{k}}$ with delay $\tilde{k}$ is non-zero if $u_i[k-\tilde{k}] \in \mathcal{S}_t$, and column $i \in [1; N_y]$ of $W^{\mathrm{h}}_{y,\tilde{k}}$ with delay $\tilde{k}$ is non-zero if $y_i[k-\tilde{k}] \in \mathcal{S}_t$ – see (4) for an example. The weights W^{o}, θ^{o} and θ^{h} are always fully populated.

4.3 Evaluation and Prediction Forecasting

When forecasting future outputs from sample p to sample k at trial t, then future inputs $u[k \mid t]$ are required due to the dynamic model structure – as briefly discussed in (4.1). To alleviate this, the known "future" inputs from the past batch $t-1$ is used as substitute for the unknown future inputs for batch t by modifying (1) according to

$$\hat{y}^*[k+1 \mid t, p, \mathcal{W}] = \begin{cases} \mathcal{N}\big(\hat{Y}^*[k], U^*[k] \mid \mathcal{W}\big), & p \le k \\ y[k+1 \mid t], & k < p \end{cases} \tag{30a}$$

$$u^*[k \mid t] = \begin{cases} u[k \mid t-1] & p < k \\ u[k \mid t] & k \le p \end{cases} \tag{30b}$$

with

$$U^*[k] = \begin{bmatrix} u^*[k-\bar{n}_1 \mid t]^T & \cdots & u^*[k-\bar{n}_{N_{\bar{n}}} \mid t]^T \end{bmatrix}^T \text{ and}$$
$$\hat{Y}^*[k] = \begin{bmatrix} \hat{y}^*[k-\bar{m}_1 \mid t,p,\mathcal{W}]^T & \cdots & \hat{y}^*[k-\bar{m}_{N_{\bar{m}}} \mid t,p,\mathcal{W}]^T \end{bmatrix}^T$$

where $\hat{y}^*[k+1 \mid t,p,\mathcal{W}]$ and $u^*[k \mid t]$ is the prediction forecasting output and input. When forecasting using (30), i.e. $\hat{y}^*[k+1 \mid t,p,\mathcal{W}]$, we denote it *prediction forecasting*, and when using (1), i.e. $\hat{y}[k+1 \mid t,p,\mathcal{W}]$, we denote it *evaluation forecasting*. The key difference between (1) and (30) is the use of $u[k \mid t-1]$ in (30b), as it ensures that the unavailable $u[k \mid t]$ for $p < k$ is not used for forecasting – this is also explained on Table 2. Note that only (1) is used for training, and that both evaluation and prediction forecasts uses the exact same model weights $\mathcal{W}$.

	Past ($k \le p$)	Future ($p < k$)
U	t	t
Y	t	$\approx t$

(a) Evaluation

	Past ($k \le p$)	Future ($p < k$)
U	t	$t-1$
Y	t	$\approx t$

(b) Prediction

Table 2: The objective is to forecast future outputs Y for $p < k$ at trial t – denoted by $\approx t$. Evaluation forecasting (a) unrealistically relies on future inputs U for $p < k$ at trial t, while prediction forecasting (b) realistically relies on past inputs U for $p < k$ at trial $t-1$. The difference between the two methods has been highlighted with a gray box. Note that both forecasting approaches rely on past inputs and outputs for initialization, i.e., U and Y for $k \le p$.

4.4 Ensemble Evaluation and Prediction Forecasting

We generate ensemble forecasts with N_m sub-models trained with different initial weights on the same training data denoted $\{\mathcal{W}_1(\mathcal{B}_t, t\text{–}2), \ldots, \mathcal{W}_{N_m}(\mathcal{B}_t, t\text{–}2)\}$. The ensemble evaluation and prediction forecasting mean of output index $l \in \mathbb{Z}_+$ are respectively given by

$$\hat{\bar{y}}_l[k \mid t,p] = \frac{1}{N_m} \sum_{i=1}^{N_m} \hat{y}_l[k \mid t,p,\mathcal{W}_i(\mathcal{B}_t, t\text{–}2)] \text{ and} \tag{32a}$$
$$\hat{\bar{y}}^*_l[k \mid t,p] = \frac{1}{N_m} \sum_{i=1}^{N_m} \hat{y}^*_l[k \mid t,p,\mathcal{W}_i(\mathcal{B}_t, t\text{–}2)], \tag{32b}$$

where $\mathcal{W}_i(\mathcal{B}_t, t\text{–}2)$ is the i'th sub-model weight with batch indexes $\mathcal{B}_t$ and early stopping applied on batch t–2. We use the ensemble model mean to represent the "true" model output, as it is expected to be significantly more robust against initialization effects.

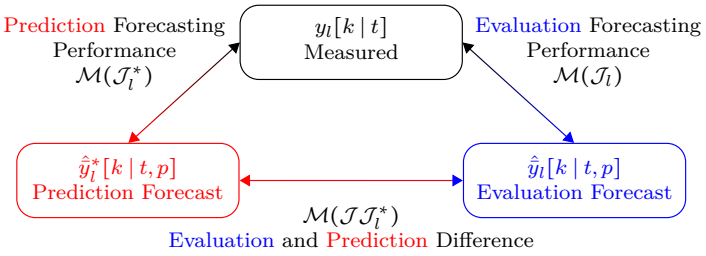


Figure 7: Overview of how (35) is related to (33) and (34).

4.5 Forecasting Evaluation Criteria

To evaluate the ensemble models' forecasting ability throughout trial t, the forecasting evaluation and prediction root mean square error (RMSE) is calculated for trial t and output l by:

$$\mathcal{J}_l(t) = \sqrt{\sum_{p=1}^{N_{s,t-1}-1} \sum_{k=p+1}^{N_{s,t-1}} \frac{\left\| y_l[k \mid t] - \hat{\hat{y}}_l[k \mid t, p] \right\|_2^2}{\frac{1}{2} N_{s,t-1} \left(N_{s,t-1} - 1 \right)}} \tag{33a}$$

$$\mathcal{J}_l^*(t) = \sqrt{\sum_{p=1}^{N_{s,t-1}-1} \sum_{k=p+1}^{N_{s,t-1}} \frac{\left\| y_l[k \mid t] - \hat{\hat{y}}_l^*[k \mid t, p] \right\|_2^2}{\frac{1}{2} N_{s,t-1} \left(N_{s,t-1} - 1 \right)}} \tag{33b}$$

Where $\mathcal{J}_l(t) \in \mathbb{R}_+$ is the evaluation forecasting performance of trial t, $\mathcal{J}_l^*(t) \in \mathbb{R}_+$ is the prediction forecasting performance of trial t, $y_l \in \mathbb{R}$ is the l'th index of $y \in \mathbb{R}^{N_y}$, and the factor $\frac{1}{2} N_{s,t-1} \left(N_{s,t-1} - 1 \right)$ is a normalization constant. Furthermore, the impact of using substitute future inputs can be quantified by the difference between $\hat{\hat{y}}_l[k \mid t, p]$ and $\hat{\hat{y}}_l^*[k \mid t, p]$, according to

$$\mathcal{J}\mathcal{J}_l^*(t) = \sqrt{\sum_{p=1}^{N_{s,t-1}-1} \sum_{k=p+1}^{N_{s,t-1}} \frac{\left\| \hat{\hat{y}}_l[k \mid t, p] - \hat{\hat{y}}_l^*[k \mid t, p] \right\|_2^2}{\frac{1}{2} N_{s,t-1} \left(N_{s,t-1} - 1 \right)}}. \tag{34}$$

The mean RMSE performance of all trials $t \in \{1, \ldots, N_t\}$, as depicted on Figure 7, is then calculated according to

$$\mathcal{M}(\mathcal{J}_l) = \frac{1}{N_t} \sum_{t=1}^{N_t} \mathcal{J}_l(t), \tag{35a}$$

$$\mathcal{M}(\mathcal{J}_l^*) = \frac{1}{N_t} \sum_{t=1}^{N_t} \mathcal{J}_l^*(t) \text{ and} \tag{35b}$$

$$\mathcal{M}(\mathcal{J}\mathcal{J}_l^*) = \frac{1}{N_t} \sum_{t=1}^{N_t} \mathcal{J}\mathcal{J}_l^*(t), \tag{35c}$$

where $\mathcal{M}(\mathcal{J}_l)$, $\mathcal{M}(\mathcal{J}_l^*) \in \mathbb{R}_+$ are the mean evaluation and prediction RMSE performance, and $\mathcal{M}(\mathcal{J}\mathcal{J}_l^*) \in \mathbb{R}_+$ is the RMS evaluation and prediction forecasting difference.

4.6 Forecasting Procedure Summary

Forecasting trial t is carried out as follows:

1) Preselect up to N_B batches according to (25) from the prior batches $\mathcal{B}_t \subseteq \{1\text{–}N_{PB}, \ldots, t\text{–}3\}$.

2) Formulate $\tilde{u}$ from $\mathcal{B}_t$ and X from $\tilde{u}$ and $\tilde{y}$. From each of element in $\tilde{y}$ carry out an IVS as described in Section 3 to calculate $\mathcal{S}_t$ using (27).

3) From $\mathcal{S}_t$ determine the MLP model $\mathcal{N}$ structure governed by $\{\bar{n}, W_{u,i}^h, \bar{m}, W_{y,j}^h\}$ as described in Section 4.2, and the inputs u and outputs y using (28).

4) Train N_m sub-model weights $\{\mathcal{W}_1, \ldots, \mathcal{W}_{N_m}\}$ with different initial conditions by minimizing (5) as described in Section 2.2.

5) Calculate the ensemble prediction forecast $\hat{\tilde{y}}_l^*[k \mid t, p]$ and evaluation forecast $\hat{\tilde{y}}_l[k \mid t, p]$ for all initial samples $p \in [1; N_{s,t-1} - 1]$ and $k \in [p+1; N_{s,t-1}]$ using (32).

6) Calculate the mean prediction and evaluation forecasting RMSE $\mathcal{M}(\mathcal{J}_l)$ and $\mathcal{M}(\mathcal{J}_l^*)$ given by (33a), and the RMS evaluation and prediction forecasting difference.

A visual overview of prediction forecasting is depicted on Figure 8.

5 Experimental Forecasting: Description

This work is based on data gathered from a ≈20 year old state-of-the-art broiler house located in the northern Denmark – not the same as used in [Johansen et al., 2017]. We use data from 29 batches, collected over a period of 3 years and 4 months. Each batch contains 38155 broilers on average with standard deviation 2329.

The potential inputs to the model consist of the available environmental variables for this broiler house: The measured inside temperature, outside temperature and humidity, light intensity reference, ventilation demand, and heating demand. Note that measured CO_2 was not included, as it was only present in 8 of 29 batches. The model outputs consist of the available broiler behavior indicators: the measured weight along with feed and water consumption per bird.

We intentionally distinguish between reference, demand and measured variables. Reference variables are independent, demand variables are determined

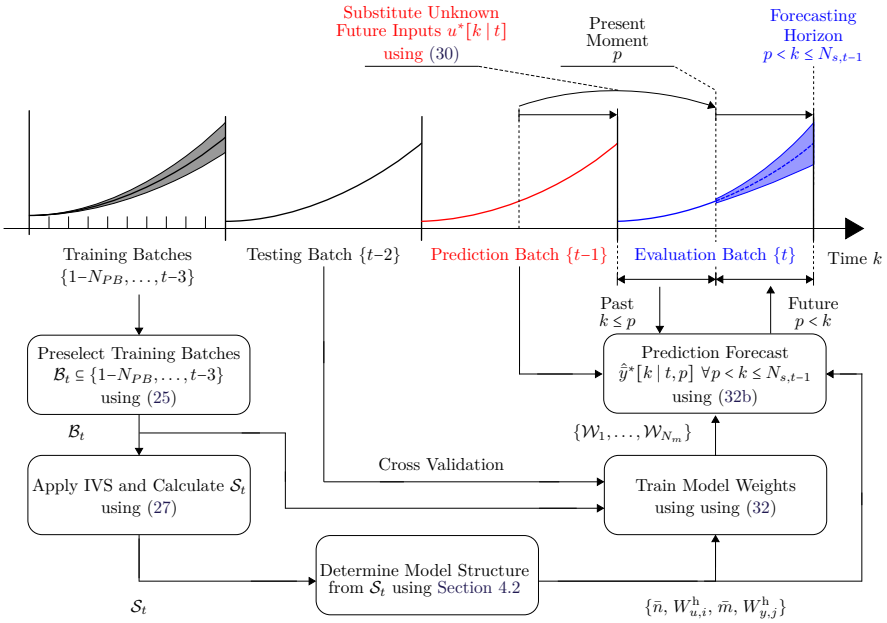


Figure 8: Visual overview of prediction forecasting. For evaluation forecasting replace $\hat{\hat{y}}^*[k \mid t, p]$ from (32b) with $\hat{\hat{y}}[k \mid t, p]$ from (32a).

by a deterministic entity like a controller outside the model, and measured variables are dependent on the model process.

To elaborate, humidity and temperature measurements are measured during closed loop operation and are controlled through both the ventilation and heating demands. Note that temperature is tightly controlled compared to humidity for safety reasons. For this reason, these variables are not independent, which has been known to cause problems in closed loop identification [Rajamani Doraiswami and Stevenson, 2014]. We do not consider this an issue as all closed loop controlled variables are considered inputs to the model and therefore no deterministic output feedback is present. Note also that reference inputs for controlled variables is determined by the broiler farmer, whom we consider a non-deterministic entity, making it an open loop identification problem with restricted input space. We give an example of this in Section 6.1.

5.1 Weight On Day Forecasting

Currently, the broiler industry is particularly attuned to the broiler weight on particular samples, e.g. samples equivalent of day 7, 14, 21 and 28, which among others are used to determine when a batch is expected to be ready

for slaughter. Therefore the models' ability to forecast the output at these days from all prior samples is investigated, which we denote *weight on day forecasting*. The model output sample k is fixed, e.g. k = day 28, and the forecasting horizon is determined through the variable initialization sample p, i.e. p = day 7, for $p < k$. Note that this differs from traditional forecasting, where p is fixed and k is variable.

5.2 Experimental Forecasting Overview

A detailed forecasting case study of the most recent trial, $t = N_t$, is presented in Section 6. The data used for modeling is analyzed to help explain when the forecasts deviate from future measured values. An example of a prediction and evaluation forecast from day 14 is presented, along with a prediction and evaluation weight on day forecast as described in Section 5.1 – both to demonstrate the forecasting application aspects.

An overview of all N_t trials are presented in Section 7. A summary of the selected inputs and outputs from the IVS method is presented and trends are identified. The trial prediction and evaluation RMS forecasting error is visualized and analyzed for all model outputs to investigate the performance of the proposed method.

5.3 Method Configuration

We use a sample interval of 3 hours with the first sample at midnight (00:00) on the first day regardless of the actual start time of the batch due to technical limitations, which permits the time of day to be partially captured. The mean value of the data within each sample period is used.

We have good experience with $N_m = 64$ sub-models and $N_h = 7$ hidden neurons in the hidden layer of the model. On average this results in 230 free model parameters across all trials. According to the Gaussian Regularization, this results in an average of $\gamma = 21$ (8.5% of the model weights) unused weights across all trials. We find this reasonable, as the model has room to handle additional complexity. We have good experience with $N_P = 5$ sub-batch training start times at day $P = \{2, 7, 14, 21, 28\}$ in (5).

As we cannot guarantee the data-quality in broiler production, we deliberately limit the amount of selected variables and number of lags – see Section 4.2 for more information. A maximum of $N_B = 10$ training and $N_{PB} = 11$ preliminary batches are used – resulting in a total of $N_t = 18$ trials. As each batch roughly contains 250 samples, Table 1 suggests that no more than 5 inputs should be selected. As the IVS is independently carried out for each of the outputs, a maximum of 18 lagged inputs can be selected. We consider input and output lags up to $N_{L,\tilde{u}} = N_{L,\tilde{y}} = 16$ samples – equivalent of 2 days. Furthermore, we restrict the search space of the IVS, S, to 3 input classes (e.g. measured temperature) per output where each input class $\tilde{u}$ can have up to 2 lags.

6 Experimental Forecasting: Trial Case Study

This sections describes a detailed case study of the most recent trial denoted by t=18 – see Section 5.2 for more information.

6.1 Trial Data Analysis

To understand the trial results we give our interpretation of the input and output variables, which are depicted on Figure 9. We emphasize the difference between what the model "knows" through training data, and the "unknown" validation and prediction data, which can help explain if the model has difficulties at generalizing. We also point out known sources of error related to broiler production that can influence the model's forecasting ability. The IVS results for this trial are depicted on Table 3 with rounded rectangles, which we will not comment on in the following.

Input variables

First we give an example of the aforementioned correlated input variables. It is most clearly illustrated on the prediction batch on Figure 9, where the ventilation demand is negatively correlated with the temperature and humidity. The intuition is that the broilers introduce both heat and water into the air in the broiler house, increasing the temperature and humidity, which are removed through ventilation.

A rule of thumb by the hatchery supplying the day old broilers is to keep the temperature at ≈30 °C in the beginning of the batch and decrement it by 1 °C every 3 days until 20 °C. The beginning temperature appears to be ≈4 °C offset in this case. We note a low degree of variation in temperature across all batches, compared to e.g. humidity. The overall measured temperature of the prediction- and evaluation batch appear to have similar behavior as the training and testing data. However, the prediction batch is on average ≈1 °C higher onwards of day ≈11.

The heating demand for the prediction and evaluation batch have significantly higher fluctuations onwards of day 15, which are caused by a below average outside temperature. Lastly, we note that the light intensity reference data are quite spiky, which is caused by a light program emulating the day and night cycle.

Output variables

Broiler weight is highly positively correlated with cumulative feed and water consumption, as the two are necessary for both survival and growth of the broilers. Bird weight has been reported by customers to be highly predictable from feed and water consumption – supported by the fact that the three have

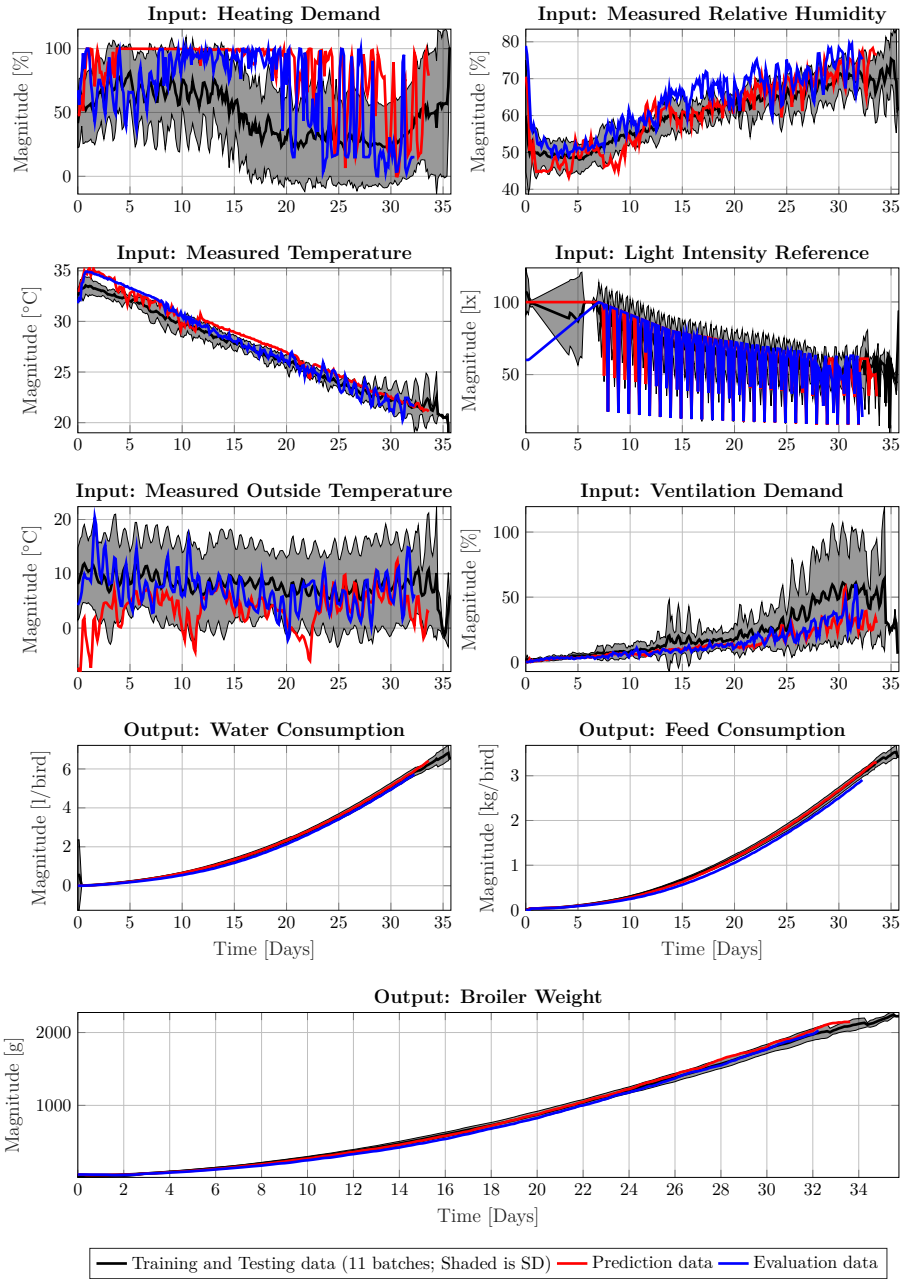


Figure 9: The inputs and output data from the batches used in the most recent trial $t = N_t$. The black shaded area denotes one standard deviation from the mean value. Each batch contains 38155 broilers on average with standard deviation 2329.

similar variations and behavior. Note that it is unknown if and when the farmer has changed feed type for any of the batches.

We note that the feed consumption is distinctly lower for the evaluation batch compared to the training, testing and prediction data, but the measured broiler weight and water consumption are within one standard deviation of the training and testing data. For this reason, we expect the prediction forecast to be higher than the evaluation forecast.

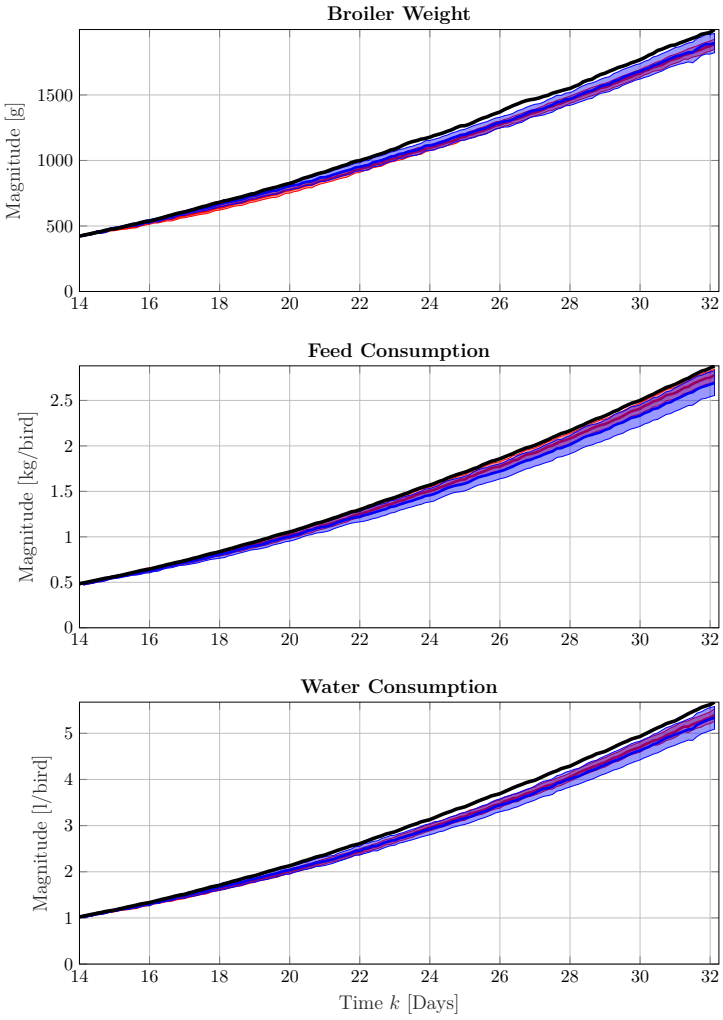
It is a common problem within broiler production that the measured weight is negatively biased from day 15 and onwards. This problem was first reported by [Newberry et al., 1985], but has subsequently received limited academic attention. [Chedad et al., 2000] hypothesized that the automatic weighting system was used less frequently by heavier broilers through image analysis, which was subsequently confirmed in [Chedad et al., 2003]. This has been corrected by multiplying a manually configured and time varying "behavior factor" with the measured weight, e.g., 1.1 at day 34. We only have the corrected weight available, which naturally leads to some uncertainty.

6.2 Discussion of Figure 10a

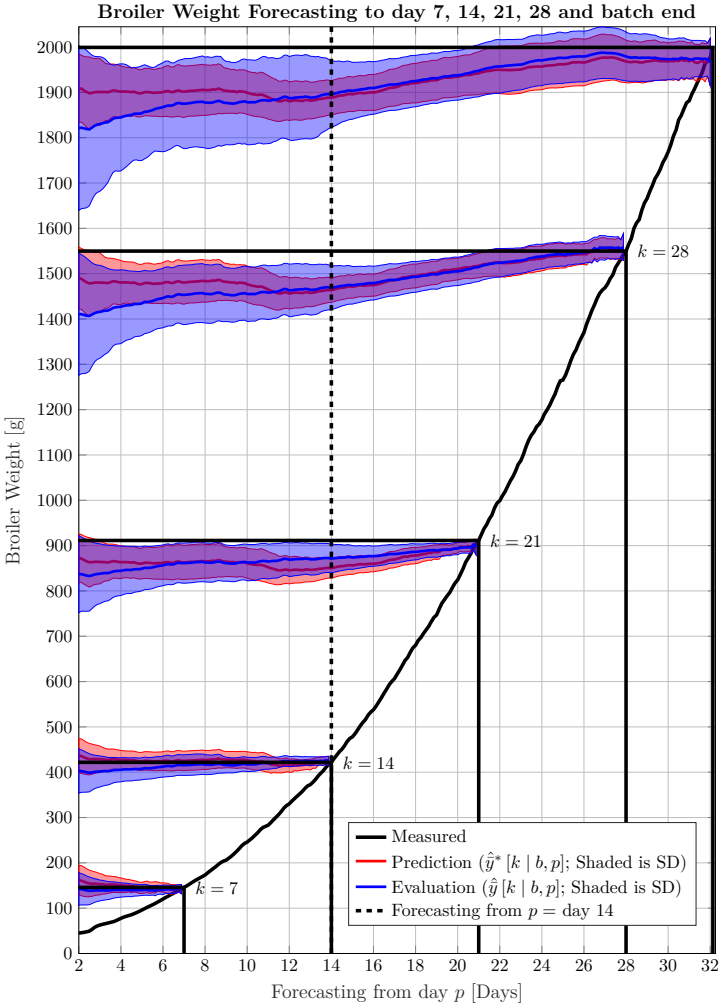
We see that both the prediction and evaluation forecasts show good overall forecasting capability, in the sense that it tracks the measured output shape throughout the batch – however, the model has a tendency to underestimate for all outputs. Comparing the prediction and evaluation forecasts, $\hat{\hat{y}}_l[k \mid t, p] - \hat{\hat{y}}_l^*[k \mid t, p]$, we note a difference of −8g broiler weight, 87g feed and 62ml water – essentially predicting a similar broiler weight for less feed and water. Comparing this with the measured data of the prediction- and evaluation batch on Figure 9 at the end of the evaluation batch, we see that the evaluation batch produces the same amount of meat with 3.8% less feed and 1.1% less water, which supports the previous observation. This indicates that the model captures at least some of the dynamic interconnection between the two forecasts, as the climate conditions are the only difference between the prediction and evaluation forecasts.

6.3 Discussion of Figure 10b

We see that the under-estimation trend is present when forecasting weight after day 14, as the weight on day 7 and 14 forecasts show good agreement with the measured weight. The weight on day 21 and 28 forecasts are underestimated when forecasting from before day ≈22, which is caused by a higher than average growth in this period. To elaborate, between day ≈7 to ≈24 the broiler weight is more than one standard deviation lower than the training and testing data, but it is close to the mean broiler weight after day ≈24 – the ensemble model cannot explain this from the input data. This could be caused by biased broiler weight measurements in this interval.



(a) Forecasting from p = day 14 to batch end. See Section 4.3 for details.



(b) Broiler Weight on day forecasting to k = day 7, 14, 21, 28 and batch end. See Section 5.1 for details.

Figure 10: The solid blue evaluation forecasts are generated with models initialized with evaluation batch data and future inputs from the evaluation batch. The solid red prediction forecasts are initialized with evaluation batch data, but with substitute future inputs from the prediction batch. The blue traces use future input data from the current batch, and the red traces use substitute future input data from the previous batch. The blue and red bold lines are the ensemble model means, while the shaded areas denote the ensemble standard deviation. Figure 10a depicts a single forecast from day p = 14 to the batch end. Figure 10b aims at showing the weight forecast of Figure 10a for all start samples, p, on specific days, k, marked with vertical solid black lines. The vertical dashed line on Figure 10b marks the forecast on Figure 10a. The horizontal solid black lines show the measured weight for that day.

Comparing the evaluation and prediction forecasts, we see that they resemble each other well in terms of ensemble model standard deviation as the shaded area is similar for the two forecasts, indicating that substitute input data from the previous batch can be used to represent the ensemble standard deviation of the current batch despite different ensemble model means.

7 Experimental Forecasting: Trial Evaluations

This section provides an overview of all 18 trials – see Section 5.2 for more information.

7.1 Input Selection Discussion

In Table 3 a summary of all selected inputs, input lags and output lags is depicted. In the following, we discuss the selection trends for each output.

Y		Relevant Potential Input Variables X	Selected Lags S (% selected of all trials)															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Broiler Weight	Input	Measured Relative Humidity	28	22	11	0	17	0	6	22	28	11	17	17	6	6	0	0
		Measured Temperature	0	0	6	6	11	6	6	6	0	0	0	0	0	6	0	0
		Light Intensity Reference	22	0	0	0	0	6	28	0	0	0	0	0	0	0	0	0
		Measured Outside Temperature	6	11	0	11	11	11	17	6	6	11	6	6	0	11	11	11
		Ventilation Demand	17	6	11	22	11	0	11	22	17	0	6	17	6	0	22	11
	Output	Broiler Weight	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Feed Consumption	39	11	6	0	0	6	0	11	11	0	22	11	39	11	17	17
		Water Consumption	44	6	6	0	0	0	11	28	6	28	17	0	22	11	6	17
Feed Consumption	Input	Heating Demand	0	0	22	6	0	0	6	6	6	0	0	17	0	0	0	28
		Measured Relative Humidity	11	11	22	11	6	33	11	17	11	0	11	11	0	0	0	0
		Measured Temperature	11	6	22	6	0	0	0	0	0	0	0	0	0	0	0	0
		Light Intensity Reference	11	0	0	0	0	0	0	11	6	0	0	6	11	11	6	17
		Measured Outside Temperature	0	11	22	11	0	6	11	6	0	6	22	6	11	0	11	22
		Ventilation Demand	17	0	22	0	6	6	11	0	0	0	6	6	11	0	6	0
	Output	Broiler Weight	39	28	17	0	22	0	33	0	6	0	0	0	11	11	17	17
		Feed Consumption	100	61	0	39	0	0	0	0	0	0	0	0	0	0	0	0
		Water Consumption	100	50	0	39	0	0	6	0	6	0	0	0	0	0	0	0
Water Consumption	Input	Heating Demand	0	0	6	0	0	0	6	0	0	0	0	0	0	0	0	0
		Measured Relative Humidity	6	17	6	6	0	11	11	6	0	0	6	6	6	11	0	6
		Measured Temperature	11	0	6	6	0	0	0	0	0	0	0	0	0	0	6	0
		Light Intensity Reference	44	0	6	0	0	0	0	44	11	0	0	0	0	0	0	6
		Measured Outside Temperature	0	6	6	0	0	6	6	0	0	0	6	17	6	17	17	0
		Ventilation Demand	39	0	17	0	0	11	11	11	0	6	28	6	0	0	0	6
	Output	Broiler Weight	0	6	11	0	0	28	17	17	6	22	6	6	0	22	0	39
		Feed Consumption	61	0	22	0	0	0	17	11	0	6	22	0	6	0	0	0
		Water Consumption	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3: Overview of the selected inputs, input lags and output lags for each of the N_t =18 trials. A separate IVS of the selected lags X resulting in the selected lags S is presented for each output Y. The lags used in the most recent trial is marked with a solid black bars, while our interpretation of trends are marked with shaded gray bars. Note that Light Intensity Reference was only present in 12/18 tests.

Broiler weight

The primarily selected inputs for the broiler weight are the measured relative humidity, measured outside temperature and ventilation demand. The measured relative humidity appears to have a short and long horizon with lags at 0-2 and 7-11 – equivalent of "now" and a day ago (8 samples per 24 hour). The measured outside temperature appears to have no clear trends in the selected lags, which suggest a seasonal dependence rather than a dependence on outside temperature. The ventilation demand also appears to have no clear trends, but an argument for small clusters at around lag 0, 3, 7, 11 and 14 can be made with a periodicity of ≈4 lags (12 hours).

The selected outputs for the broiler weight appear to be strongly correlated with lag 0 and 1 of itself. It also appears to be dependent on the two other outputs, either very recently at lag 0 and 1 or after lag 7. A similar, but entirely heuristic claim was made in [Johansen et al., 2017].

Feed Consumption

The primary selected inputs for the feed consumption are the measured relative humidity and measured outside temperature. The measured relative humidity appears to be selected within the first 8 lags. Like the broiler weight input selection, the measured outside temperature appears to have no clear trends.

The selected outputs for the feed consumption are primarily between lag 0 and 3 for both the feed consumption and water consumption. Broiler weight appears to have a short term component between lag 0 and 6 and a long term component between lag 12-15.

Water Consumption

The primary selected inputs for the water consumption are the ventilation demand, light intensity reference and measured relative humidity. The ventilation demand appears to have a weak trend between lag 0 and 11 – an argument can be made for small clusters at lag 0, 2, 6 and 10. The light intensity reference has a strong trend at lag 0 and 7, indicating a day rhythm as they are spaced ≈24 hours apart. The measured relative humidity appears to have no strong trends, but it is likely that there are small clusters at lag 1, 5, 12 and 15.

The selected outputs for the water consumption appear to be strongly correlated with lag 0 and 1 of itself, like the broiler weight output. Broiler weight appears to have a long term component between lag 5 and 15, possibly with a small cluster at lag 7, 13 and 15. Feed consumption appears to have two clusters, one between lag 0 and 2, and one between lag 6 to 10.

Other Trends

The measured temperature was not primarily selected for any of the outputs – but it appears to have great influence on the ensemble model presented in

Section 6. This might suggest that the temperature is in an optimum range for the majority of the batches, as we know that temperature has a strong influence on broiler growth. Alternatively, this might be due to it being tightly controlled.

Heating demand was not primarily used for any of the outputs, but comes quite close in feed consumption. Hence, heating demand might contribute little to broiler forecasting.

Disease and antibiotics are known to have a significant influence on broiler growth. Due to the random and sparsely appearing nature of these variables, it is difficult to incorporate into a dynamic model, which will influence the forecasting performance. Note that other tools, such as fault detection algorithms, are better at dealing with such information.

7.2 Trial Performance Evaluations

In this section we comment on the RMS forecasting error depicted on Figure 11 and Table 4 From Figure 11 we see no pattern for broiler weight and feed consumption, but an argument could be made for a seasonal influence for the water consumption with a period of ≈6 trials – equivalent to ≈1 year. This is not a source of concern, as the main objective is broiler weight forecasting. Most importantly, $\mathcal{J}^*$ appears to consistently track the trend of $\mathcal{J}$, suggesting that substituting unknown future inputs with inputs from a previous batch does not degrade performance significantly. From the mean RMS forecasting errors from Table 4, we see that $\mathcal{M}(\mathcal{J}\mathcal{J}^*)\neq 0$ and $\mathcal{M}(\mathcal{J}^*)\approx\mathcal{M}(\mathcal{J})$ for all outputs – see Figure 7 for clarification. *This strongly suggests that using substitute future inputs in forecasting does not deteriorate forecasting performance significantly, while simultaneously suggesting that the broiler house conditions have a significant impact on the outputs*, e.g., a RMSE difference of 2.7g in the case of broiler weight forecasting. In conclusion, the method is shown to be practically applicable to broiler weight forecasting and the achieved mean forecasting RMSE of 66.8g is acceptable in state-of-the-art broiler industry.

Output	Mean Forecasting RMSE		
	$\mathcal{M}(\mathcal{J}^*)$	$\mathcal{M}(\mathcal{J})$	$\mathcal{M}(\mathcal{J}\mathcal{J}^*)$
Weight [g]	66.8	64.1	35.5
Feed [g/bird]	90.2	87.8	63.6
Water [ml/bird]	185.8	169.2	133.2

Table 4: The mean prediction and evaluation forecasting RMSE along with their difference are calculated using (35).

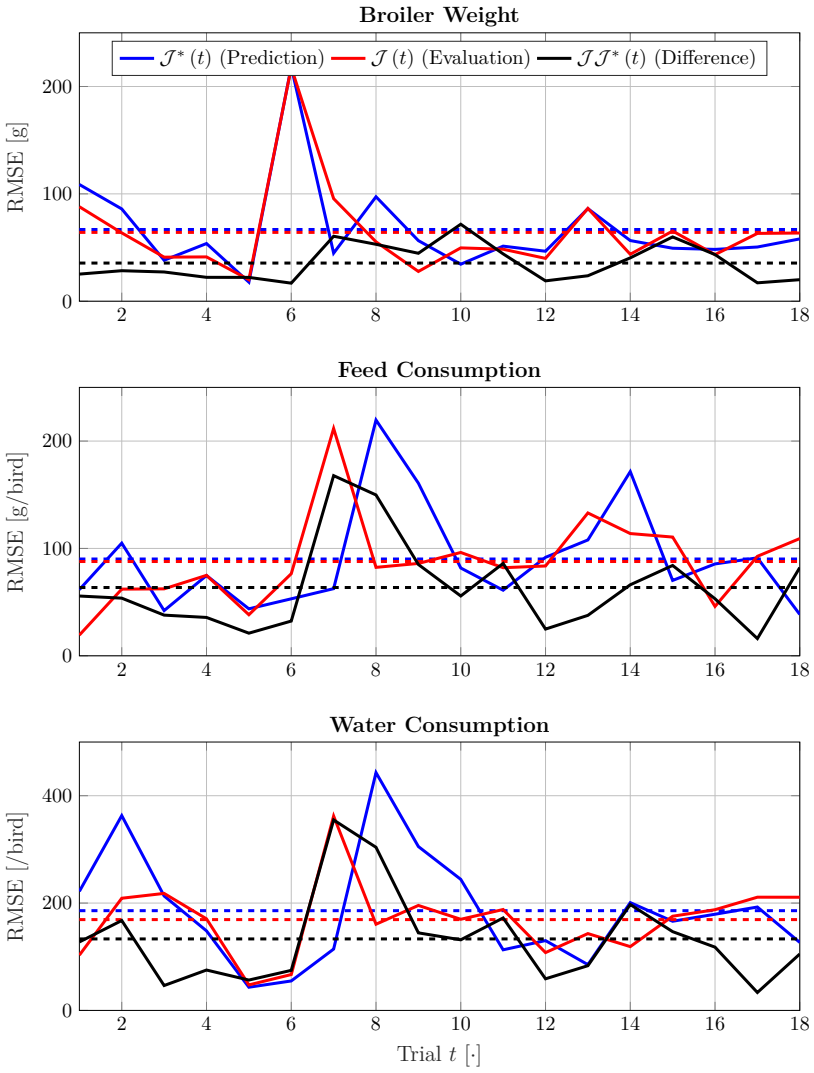


Figure 11: Prediction and evaluation RMS forecasting error calculated using (33) and (34) of all trials, where a dashed line is the mean from Table 4.

8 Concluding Remarks

In this work the first results on broiler weight forecasting intended for production planning and monitoring by means of environmental broiler house conditions have been presented using dynamic neural network models with mutual information based input variable selection. The method only uses past production data, and an extensive case study on almost 3.5 years of production data was conducted to verify the validity of the proposed method. The dynamic impact of environmental conditions on broiler growth is investigated and found to be significant, which was previously known to be highly significant but unexplored in scientific literature. The mean forecasting broiler weight root mean square error across 18 forecasting trials was found to be 66.8g, making the method potentially useful in industry. This effectively provides a foundation for future research on optimization of broiler production.

Future work on broiler weight forecasting includes improving the mean forecasting root mean square error by exploring state estimation, modifying the method to better cope with biased broiler weight measurements, and determining the optimal number of neurons. Lastly, a forecasting confidence measure based on how much the model has to generalize from known training data to prediction data would help assess the forecasting quality.

Acknowledgments

The authors would like to thank broiler manager Erik Hjorth for permitting access to the data used in this work. This work was supported by Innovationsfonden [grant number 5016-00133B].

References

J. M. Aerts, M. Lippens, G. De Groote, J. Buyse, E. Decuypere, E. Vranken, and D. Berckmans. “Recursive prediction of broiler growth response to feed intake by using a time-variant parameter estimation method”. *Poultry Science*, 82:40–49, 2003. ISSN 0032-5791.

S. Aggrey. “Comparison of three nonlinear and spline regression models for describing chicken growth curves”. *Poultry Science*, 81(12):1782–1788, dec 2002. doi:10.1093/ps/81.12.1782.

H. Ahmadi and M. Mottaghitalab. “Hyperbolastic models as a new powerful tool to describe broiler growth kinetics”. *Poultry Science*, 86(11):2461–2465, nov 2007. doi:10.3382/ps.2007-00086.

S. Arlot and A. Celisse. “A survey of cross-validation procedures for model selection”. *Statistics Surveys*, 4(0):40–79, 2010. doi:10.1214/09-ss054.

J. Beirlant, E. J. Dudewicz, L. Györfi, and E. C. Van Der Meulen. "Nonparametric entropy estimation: An overview". *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.5281&rep=rep1&type=pdf.

G. J. Bowden, H. R. Maier, and G. C. Dandy. "Input determination for neural network models in water resources applications. part 2. case study: forecasting salinity in a river". *Journal of Hydrology*, 301(1-4):93–107, jan 2005. doi:10.1016/j.jhydrol.2004.06.020.

G. Brown, A. Pocock, M. jie Zhao, M. Luján, and I. Guyon. "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection". *Journal of Machine Learning Research*, 13, 2012.

F. Burden and D. Winkler. "Bayesian regularization of neural networks". In *Methods in Molecular Biology*, pages 23–42. Springer Science + Business Media, 2008. doi:10.1007/978-1-60327-101-1_3.

O. Cangar, J.-M. Aerts, E. Vranken, and D. Berckmans. "Online growth control as an advance in broiler farm management". *Poultry Science*, 86:439–443, 2007. ISSN 0032-5791.

A. Chedad, E. Vranken, J.-M. Aerts, and D. Berckmans. "Behaviour of chickens towards automatic weighing systems". *IFAC Proceedings Volumes*, 33(29):207 – 212, 2000. doi:https://doi.org/10.1016/S1474-6670(17)36778-2.

A. Chedad, J.-M. Aerts, E. Vranken, M. Lippens, J. Zoons, and D. Berckmans. "Do heavy broiler chickens visit automatic weighing systems less than lighter birds?". *British Poultry Science*, 44(5):663–668, dec 2003. doi:10.1080/00071660310001643633.

L. Chen, L. Ye, V. Singh, J. Zhou, and S. Guo. "Determination of input for artificial neural networks for flood forecasting using the copula entropy method". *Journal of Hydrologic Engineering*, 19(11):04014021, nov 2014. doi:10.1061/(asce)he.1943-5584.0000932.

T. Demmers, Y. Cao, S. Gauss, J. Lowe, D. Parsons, and C. Wathes. "Neural predictive control of broiler chicken growth". *IFAC Proceedings Volumes*, 43(6):311–316, 2010. doi:10.3182/20100707-3-be-2012.0061.

L. F. Demuner, D. Suckeveris, J. A. Muñoz, V. C. Caetano, C. G. de Lima, D. E. de Faria Filho, and D. E. de Faria. "Adjustment of growth models in broiler chickens". *Pesquisa Agropecuária Brasileira*, 52(12):1241–1252, dec 2017. doi:10.1590/s0100-204x2017001200013.

A. Diez-Olivan, X. Averós, R. Sanz, B. Sierra, and I. Estevez. "Quantile regression forests-based modeling and environmental indicators for decision support in broiler farming". *Computers and Electronics in Agriculture*, mar 2018. doi:10.1016/j.compag.2018.03.025. in press.

K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer Science + Business Media, 2014. doi:10.1007/978-1-4471-5571-3.

J. Grahovac, A. Jokić, J. Dodić, D. Vučurović, and S. Dodić. "Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks". *Renewable Energy*, 85:953–958, jan 2016. doi:10.1016/j.renene.2015.07.054.

A. e. F. N. o. Hasan Eleroğlu, Arda Yıldırım and M. Duman. "Comparison of growth curves by growth models in slow–growing chicken genotypes raised the organic system". *International Journal of Agriculture and Biology*, 16(3):529–535, 2014.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.

J. He, C. Valeo, A. Chu, and N. F. Neumann. "Prediction of event-based stormwater runoff quantity and quality by ANNs developed using PMI-based input selection". *Journal of Hydrology*, 400(1-2):10–23, mar 2011. doi:10.1016/j.jhydrol.2011.01.024.

S. C. Hernandez, J. A. Bueno, E. N. Sanchez, and L. Diaz-Jimenez. "State estimation by artificial neural networks in a continuous bioreactor". *IFAC Proceedings Volumes*, 46(31):215–220, 2013. doi:10.3182/20131216-3-in-2044.00033.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

X. Li, H. R. Maier, and A. C. Zecchin. "Improved PMI-based input variable selection approach for artificial neural network and other data driven environmental and water resource models". *Environmental Modelling & Software*, 65:15–29, mar 2015. doi:10.1016/j.envsoft.2014.11.028.

A. Z. Lopes, L. Ferreira, T. Y. Junior, and W. S. Lacerda. "Modeling productive performance of broiler chickens with artificial neural network". In *Livestock Environment VIII, 31 August - 4 September 2008, Iguassu Falls, Brazil*. American Society of Agricultural and Biological Engineers (ASABE), 2008. doi:10.13031/2013.25602.

K. Madsen, H. B. Nielsen, and O. Tingleff. "Methods for non-linear least squares problems (2nd ed.)". Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004.

R. May, G. Dandy, and H. Maier. "Review of input variable selection methods for artificial neural networks". In *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech, apr 2011. doi:10.5772/16004.

R. J. May, G. C. Dandy, H. R. Maier, and J. B. Nixon. "Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems". *Environmental Modelling & Software*, 23(10-11):1289–1299, oct 2008a. doi:10.1016/j.envsoft.2008.03.008.

R. J. May, H. R. Maier, G. C. Dandy, and T. G. Fernando. "Non-linear variable selection for artificial neural networks using partial mutual information". *Environmental Modelling & Software*, 23(10-11):1312–1326, oct 2008b. doi:10.1016/j.envsoft.2008.03.007.

V. V. Nair, H. Dhar, S. Kumar, A. K. Thalla, S. Mukherjee, and J. W. Wong. "Artificial neural network based modeling to evaluate methane yield from biogas in a laboratory-scale anaerobic bioreactor". *Bioresource Technology*, 217:90–99, oct 2016. doi:10.1016/j.biortech.2016.03.046.

R. Nasimi and R. Irani. "Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks". *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14):1604–1611, may 2014. doi:10.1080/15567036.2011.592903.

R. C. Newberry, J. R. Hunt, and E. E. Gardiner. "Behaviour of roaster chickens towards an automatic weighing perch". *British Poultry Science*, 26(2):229–237, apr 1985. doi:10.1080/00071668508416808.

D. Nguyen and B. Widrow. "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights". In *1990 IJCNN International Joint Conference on Neural Networks*. Institute of Electrical & Electronics Engineers (IEEE), 1990. doi:10.1109/ijcnn.1990.137819.

OECD. *OECD-FAO Agricultural Outlook 2018-2027*. OECD Publishing, jul 2018. doi:10.1787/agr_outlook-2018-en.

A. Overgaard, C. S. Kallesoe, J. D. Bendtsen, and B. K. Nielsen. "Input selection for return temperature estimation in mixing loops using partial mutual information with flow variable delay". In *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, aug 2017. doi:10.1109/ccta.2017.8062650.

C. D. Rajamani Doraiswami and M. Stevenson. "Closed loop identification". In *Identification of Physical Systems*, pages 357–378. Wiley-Blackwell, mar 2014. doi:10.1002/9781118536483.ch8.

M. Sadeghassadi, C. J. Macnab, B. Gopaluni, and D. Westwick. "Application of neural networks for optimal-setpoint design and MPC control in biological wastewater treatment". *Computers & Chemical Engineering*, 115:150–160, jul 2018. doi:10.1016/j.compchemeng.2018.04.007.

K. M. Sameer Agarwal and Others. "Ceres solver". `http://ceres-solver.org`, 2015.

A. Sharma. "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — a strategy for system predictor identification". *Journal of Hydrology*, 239(1-4):232–239, dec 2000. doi:10.1016/s0022-1694(00)00346-2.

K. Stacey, D. Parsons, A. Frost, C. Fisher, D. Filmer, and A. Fothergill. "An automatic growth and nutrition control system for broiler production". *Biosystems Engineering*, 89(3):363–371, nov 2004. doi:10.1016/j.biosystemseng.2004.07.006.

H. Wang, X. Yan, H. Chen, C. Chen, and M. Guo. "Chlorophyll-a predicting model based on dynamic neural network". *Applied Artificial Intelligence*, 29(10):962–978, nov 2015. doi:10.1080/08839514.2015.1097142.

L. Wang, Y. Yan, X. Wang, and T. Wang. "Input variable selection for data-driven models of coriolis flowmeters for two-phase flow measurement". *Measurement Science and Technology*, 28(3), 2017. URL: `http://stacks.iop.org/0957-0233/28/i=3/a=035305`.

S. S. Wolfgang Härdle, Marlene Müller and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer, 2004.

Paper C

Broiler Slaughter Weight Forecasting using Dynamic Neural Network Models

Simon V. Johansen[a,b], Jan D. Bendtsen[b] and Jesper Mogensen[b]

[a]SKOV A/S, Hedelund 4, Glyngøre, Denmark
[b]Department of Control and Automation, Aalborg University, Denmark

Abstract—*The global broiler meat demand is expected to steadily increase, which motivates the need for better broiler production monitoring, planning, and optimization tools. This paper is concerned with negatively biased broiler weight measurements onwards of day 15-21, which is a known but unsolved problem in* state-of-the-art *industrial broiler production. By introducing and emphasizing the accurately measured broiler slaughter weight during training of a dynamic neural network, the root mean squared slaughter weight forecasting error was reduced from 162.4g to 65.4g on more than 4.5 years of* state-of-the-art *industrial production data. This paper furthermore provides insight into effective algorithm settings.*

Keywords—*Biosystems; Neural networks; Agricultural engineering*

Published in *Proceedings of International Conference on Industrial Engineering and Applications*, 2019.

Contents

1 Introduction

Due to a growing middle class in developing countries, the global poultry meat demand is expected to increase by 18% by 2027[OECD, 2018, pp. 24, 37]. The largest increase is expected in broiler (chicken for meat production) meat, which motivates the need for better broiler production monitoring, planning, and optimization tools. This paper presents the first results on broiler slaughter weight forecasting intended for production planning and monitoring, in addition to aiding the development of future production optimization strategies.

Available dynamic broiler growth models in scientific literature are very limited and exclusively use feed uptake and composition as input variables, which is understandable as feed is considered the biggest expense in broiler production. In [Aerts et al., 2003], a time-variant parameter estimation model was used to predict broiler weight 7 days ahead without a priori information. In [Stacey et al., 2004], a dynamic model using feed uptake and composition of two feed types with known nutritional value was used to predict broiler growth. In [Demmers et al., 2010], a small differential recurrent neural network was used to model broiler growth using feed uptake. However, these models are incompatible with *state-of-the-art* broiler industry, as it uses ad libitum feeding and drinking regimes – excluding production optimization using feed uptake regulation.

A viable production optimization strategy is to regulate the production environment, such as temperature and humidity, which if done correctly, is known to minimize the production footprint in terms of feed, water, and electricity. The only compatible scientific dynamic modeling literature are [Johansen et al., 2017] and [Johansen et al., 2019], which both apply dynamic neural network models for broiler weight forecasting using environmental conditions. Those works aim at forecasting the day-to-day broiler weight, of which measurements are known to be biased after day 15-21. Although the bias mechanism is unclear, it does make it difficult to use such a model for production optimization. It introduces additional uncertainty when accessing broiler weight, of which deviations between the actual and predicted slaughter weight results in reduced meat yield, i.e., inefficient slaughter. This problem was first reported by [Newberry et al., 1985], but has subsequently received limited academic attention. [Chedad et al., 2000] hypothesized that the automatic weighting system was used less frequently by heavier broilers through image analysis, which was subsequently confirmed in [Chedad et al., 2003].

Dynamic neural networks have been successfully applied to model many other complex dynamic biological processes. Recent applications include prediction of dissolved oxygen in wastewater treatment [Sadeghassadi et al., 2018], prediction of bioethanol production in a bioreactor [Grahovac et al., 2016], bioreactor prediction [Nair et al., 2016], algae growth prediction in a laboratory setting [Wang et al., 2015], yeast fermentation modeling in a bioreactor [Nasimi and Irani, 2014] and state estimation in a continuous bio reactor [Hernandez et al., 2013].

This paper is a continuation of [Johansen et al., 2019], and its main scientific contribution is to extend the broiler weight forecasting algorithm to slaughter weight forecasting, by including the slaughter weight in the training. This improves the forecasting algorithm's slaughter weight forecasting performance and allows for further analysis of the biased measured broiler weight. In [Johansen et al., 2019] the hyper-parameters were chosen heuristically, such as number of hidden neurons and sampling interval. Thus, the secondary contribution of this paper is to provide insights into the effective hyper-parameter settings.

The remainder of the paper is structured as follows. In Section 2 the broiler slaughter weight forecasting method is described, of which the forecasting trial setup is described in Section 2.2, the nominal algorithm is described in Section 2.3, and the improved algorithm is presented in Section 2.4. In Section 3 the experimental verification is described, consisting of data, test, evaluation criteria and results. Finally, concluding remarks are provided in Section 4.

2 Method

2.1 Notation

Let $\|a\|_2 = \sqrt{a^T a}$ be the standard Euclidean 2-norm of the vector a. Let b and c be sets, then $\#b$ denotes the cardinality of b and $b \backslash c = \{x \in b \mid x \notin c\}$ is the difference of b and c.

2.2 Forecasting Trial Setup

Industrial *state-of-the-art* broiler production is carried out in batches, where a large group of day-old broilers are simultaneously matured in a controlled production environment. When the group reaches its intended weight they are simultaneously slaughtered, of which the process is repeated following a thorough cleaning. Accurate slaughter weight forecasting helps the farmer to decide when the broilers have the desired weight. Hence, the aim of forecasting trial $t \in \mathbb{Z}$ is to forecast future broiler weight throughout the evaluation batch t based on past batches denoted by $\{1-N_{PB}, \ldots, t-1\}$, where N_{PB} preliminary batches are available before the first trial $t = 1$. The batches are partitioned into three categories – namely training $\{1-N_{PB}, \ldots, t-2\}$, prediction $\{t-1\}$, and evaluation $\{t\}$ as depicted on Figure 1. The training batches are used to train a dynamic model, which is used to forecast future outputs throughout the evaluation batch t.

Due to the dynamic model structure, unknown future inputs ahead of the present moment $p \in \mathbb{Z}_+$ are required to forecast future outputs on the evaluation batch t. To alleviate this, [Johansen et al., 2019] use "future" inputs from the prediction batch $t-1$ as a substitute for unknown "future" inputs ahead of the present moment p at the evaluation batch t as illustrated on Figure 1. Using unknown "future" inputs to evaluate slaughter weight prediction is difficult, as

batches have different length and slaughter information is only known at the end of each batch.

In this study, future inputs ahead of the present moment p from the evaluation batch t is unrealistically used to forecast future outputs – hence, the prediction batch t–1 is unused.

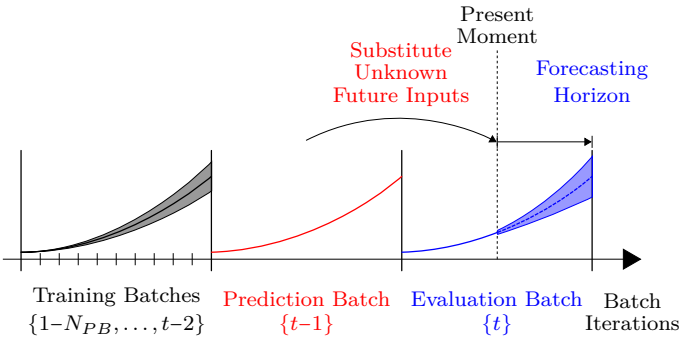


Figure 1: Illustration of forecasting and batch-segmentation for forecasting trial t with N_{PB} preliminary batches.

2.3 Nominal Forecasting Algorithm

This work extends the broiler weight forecasting algorithm presented in [Johansen et al., 2019], of which relevant details are briefly recounted.

Model

The particular type of Dynamic Neural Network (DNN) model used in this work can be classified as a discrete-time nonlinear autoregressive-moving-average model with exogenous inputs given by

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = \mathcal{N}\big(\hat{Y}[k],\, U[k] \mid \mathcal{W}\big) \qquad \forall p \le k \qquad (1a)$$

$$\hat{y}[k+1 \mid b,p,\mathcal{W}] = y[k+1 \mid b] \qquad \forall k < p \qquad (1b)$$

with

$$U[k] = \big[u[k-\bar{n}_1 \mid b]^T \;\; \cdots \;\; u[k-\bar{n}_{N_{\bar{n}}} \mid b]^T\big]^T$$

$$\hat{Y}[k] = \big[\hat{y}[k-\bar{m}_1 \mid b,p,\mathcal{W}]^T \;\; \cdots \;\; \hat{y}[k-\bar{m}_{N_{\bar{m}}} \mid b,p,\mathcal{W}]^T\big]^T$$

where $\hat{y}[k \mid b,p,\mathcal{W}] \in \mathbb{R}^{N_y}$ is the model output at sample $k \in \mathbb{Z}_+$, initialized with data from batch $b \in \mathbb{Z}$ at sample $p \in \mathbb{Z}_+$ with the $N_{\mathcal{W}}$ model weights $\mathcal{W} \in \mathbb{R}^{N_{\mathcal{W}}}$, $y[k \mid b] \in \mathbb{R}^{N_y}$ is the measured output and $u[k \mid b] \in \mathbb{R}^{N_u}$ is the measured input at batch $b \in \mathbb{Z}$ and sample $k \in \mathbb{Z}_+$, $\mathcal{N}\colon \mathbb{R}^{N_y N_{\bar{m}}} \times \mathbb{R}^{N_u N_{\bar{n}}} \to \mathbb{R}^{N_y}$ is a Multilayer perceptron (MLP) model, $U[k] \in \mathbb{R}^{N_u N_{\bar{n}}}$ is delayed values of the input vector $u[k \mid b]$ corresponding to the $N_{\bar{n}}$ elements of $\bar{n} = \{\bar{n}_1, \ldots, \bar{n}_{N_{\bar{n}}}\}$,

and $\hat{Y}[k] \in \mathbb{R}^{N_y N_{\bar{m}}}$ is delayed values of the past output vector $\hat{y}[k \mid b, p, \mathcal{W}]$ corresponding to the $N_{\bar{m}}$ elements of $\bar{m} = \{\bar{m}_1, \ldots, \bar{m}_{N_{\bar{m}}}\}$. The sample number k and initialization sample p for the model output $\hat{y}[k \mid b, p, \mathcal{W}]$ is bounded by $p \in [1; N_{s,b}]$ and $k \in [p+1; N_{s,b}]$ where $N_{s,b}$ is the number of samples in batch b. This particular structure has been adopted to accommodate potentially long propagation delays, while keeping the number of weights relatively low. Model initialization occurs through (1b), where k is implicitly lower bounded by 1 for both $y[k \mid b]$ and $u[k \mid b]$.

The DNN model $\mathcal{N}$ is chosen to have one hidden layer with hyperbolic tangent activation function in the hidden layer and linear activation function in the output layer. In matrix-vector representation, (1a) is written explicitly as

$$\hat{y}[k+1 \mid b, p, \mathcal{W}] = W^{\mathrm{o}} \tanh\left(\mathcal{X} + \theta^{\mathrm{h}}\right) + \theta^{\mathrm{o}} \tag{3}$$

with

$$\mathcal{X} = \sum_{i=1}^{N_{\bar{m}}} W^{\mathrm{h}}_{y,i} \hat{y}[k - \bar{m}_i \mid b, p, \mathcal{W}] + \sum_{j=1}^{N_{\bar{n}}} W^{\mathrm{h}}_{u,j} u[k - \bar{n}_j \mid b],$$

where $N_h \in \mathbb{Z}_+$ is the number of neurons in the hidden layer, $\mathcal{X} \in \mathbb{R}^{N_h}$, $W^{\mathrm{o}} \in \mathbb{R}^{N_y \times N_h}$ is the output weights, $W^{\mathrm{h}}_{y,i} \in \mathbb{R}^{N_h \times N_y}$ is the delayed output weights, $W^{\mathrm{h}}_{u,j} \in \mathbb{R}^{N_h \times N_u}$ is the delayed input weights, $\theta^{\mathrm{h}} \in \mathbb{R}^{N_h}$ is the hidden layer bias and $\theta^{\mathrm{o}} \in \mathbb{R}^{N_y}$ is the output bias.

Input Variable Selection (IVS)

For detailed information about the IVS algorithm see [Johansen et al., 2019]. State-of-the-art broiler production typically execute 5-8 batches per house per year. The production parameters naturally change over time as the broiler house deteriorates and both the broiler and feed performance increases. This effectively results in a parameter-drift, which drastically reduces the amount of usable production data – although the parameter-drift rate has not yet been investigated. Furthermore, data quantity requirement scales exponentially with the number of inputs, input lags and output lags for the algorithm[Johansen et al., 2019]. To alleviate this problem, mutual information based IVS is used to select the most significant inputs, input lags and output lags to make best use of the available production data.

The IVS result is included in the forecasting algorithm by modifying the structure of $W^{\mathrm{h}}_{u,j}$, $\bar{n}$, $W^{\mathrm{h}}_{y,i}$, and $\bar{m}$. For example, if the inputs indexed 1 and 3 are selected with delay of $j = 2$, $N_u = 4$ inputs, $N_h = 3$ hidden neurons, then $W^{\mathrm{h}}_{u,j}$ equals

$$W^{\mathrm{h}}_{u,2} = \begin{bmatrix} \mathcal{W}_1 & 0 & \mathcal{W}_2 & 0 \\ \mathcal{W}_3 & 0 & \mathcal{W}_4 & 0 \\ \mathcal{W}_5 & 0 & \mathcal{W}_6 & 0 \end{bmatrix}. \tag{4}$$

Keep in mind that all inputs and outputs are not guaranteed to be present in all the available batches. To maximize the amount of available information, up to $N_B \in \mathbb{Z}_+$ potential batches are selected for the IVS algorithm by maximizing

$$\begin{aligned} \mathcal{B}_t = \arg\max_{\tilde{\mathcal{B}}} \; & N_{\tilde{u}}(\tilde{\mathcal{B}}) N_{\tilde{y}}(\tilde{\mathcal{B}}) \min\{\#\tilde{B},\, N_B\} + \tfrac{\min\{\tilde{\mathcal{B}}\}}{\max\{\tilde{\mathcal{B}}\}+1} \\ \text{s.t.} \quad & \tilde{\mathcal{B}} \subseteq \{1 - N_{PB}, \ldots, t-2\} \end{aligned} \tag{5}$$

where $\mathcal{B}_t$ is the set of batches used for IVS and training of trial t, $\tilde{\mathcal{B}}$ is a set of potential batch indexes, $N_{\tilde{u}}(\tilde{\mathcal{B}})$ and $N_{\tilde{y}}(\tilde{\mathcal{B}})$ is the number of potential inputs and outputs with batch indexes $\tilde{\mathcal{B}}$, and N_B is the maximally considered batches. The term $\frac{\min\{\tilde{\mathcal{B}}\}}{\max\{\tilde{\mathcal{B}}\}+1} < 1$ ensures that the most recent batches take priority, if more than N_B batches are available, where $\min\{\tilde{\mathcal{B}}\}$ is the smallest index and $\max\{\tilde{\mathcal{B}}\}$ is the largest index.

Model Training

The model weights are found through training by solving

$$\mathcal{W}(\mathcal{B}_t) = \arg\min_{\tilde{\mathcal{W}}} \sum_{b \in \mathcal{B}_t \setminus \min\{\mathcal{B}_t\}} \frac{E_b(\tilde{\mathcal{W}})}{\#\mathcal{B}_t - 1} \tag{6}$$

with

$$E_b(\tilde{\mathcal{W}}) = \sum_{p=1}^{N_P} \sum_{k=P_p}^{N_{s,b}} \frac{\|y[k \mid b] - \hat{y}[k \mid P_p, b, \tilde{\mathcal{W}}]\|_2^2}{N_{r,b}} + \bar{\alpha}\|\tilde{\mathcal{W}}\|^2$$

$$N_{r,b} = \sum_{p=1}^{N_P} \sum_{k=P_p}^{N_{s,b}} N_y = N_P N_y (N_{s,b} + 1) - N_y \sum_{p=1}^{N_P} P_p$$

where $\mathcal{W}(\mathcal{B}_t) \in \mathbb{R}^{N_\mathcal{W}}$ is the trained weights with batch indexes in $\mathcal{B}_t$, $\tilde{\mathcal{W}} \in \mathbb{R}^{N_\mathcal{W}}$ is the potential weights, and $N_{s,b} \in \mathbb{Z}_+$ is the number of samples in batch b. The initial network weights $\tilde{\mathcal{W}}$ are initialized using the Nguyen-Widrow algorithm as explained in [Nguyen and Widrow, 1990]. (6) is minimized using the Levenberg-Marquardt algorithm with early stopping applied to the oldest batch in $\mathcal{B}_t$ – equivalent of $\min\{\mathcal{B}_t\}$. Early stopping is facilitated by selecting the model weights $\tilde{\mathcal{W}}$ among all training iterations with the smallest cost $E_{\min\{\mathcal{B}_t\}}(\tilde{\mathcal{W}})$ during training. Note that batch $\min\{\mathcal{B}_t\}$ is only used for early stopping and not directly for training in (6).

Each batch is divided into $N_P \in \mathbb{Z}_+$ Sub-Batches with initial starting times denoted by $P = \{P_1, \ldots, P_{N_P}\}$, which we find speeds up training and decreases the risk of converging to a poor local minima. Consequently, for each batch we get N_P sets of trajectories with different initial conditions to minimize, as illustrated on Figure 2, where each output is weighted equally.

The last term of the cost function, $\bar{\alpha}\|\tilde{\mathcal{W}}\|_2^2$, is a scalar regularization term punishing the size of the $N_\mathcal{W}$ system weights $\tilde{\mathcal{W}}$, where $\bar{\alpha} \in \mathbb{R}_+$ is the regularization weight. The regularization weight $\bar{\alpha}$ is determined iteratively through

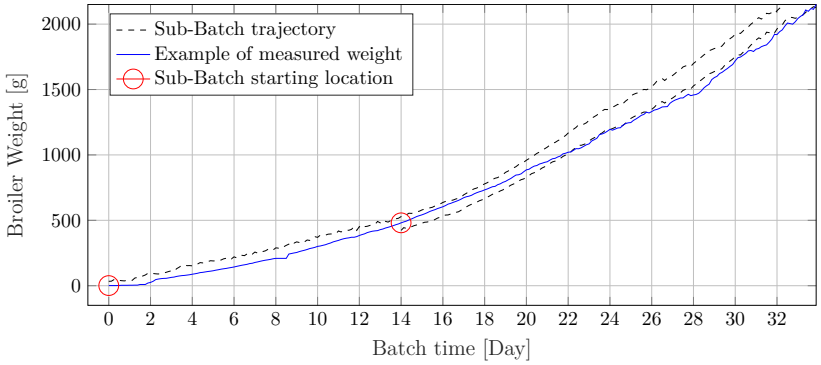


Figure 2: Example of the Sub-Batch trajectories with Sub-Batch starting time at day $P = \{0, 14\}$.

Bayesian Regulation as described in [Burden and Winkler, 2008] and [Johansen et al., 2019].

2.4 Improved Slaughter Weight Forecasting Algorithm

Automatic weighing pads are commonly used for weighing broilers, which require one or more broilers to physically stand on the pad to be measured. The measured broiler weight is known to be negatively biased onwards of day 15-21, although the exact mechanism behind this bias is not known, which has been partially corrected by introducing a heuristic time-dependent correction factor. For instance, in [Johansen et al., 2017] it was postulated that the negative bias could be caused by the broilers outgrowing their skeletal capacity, resulting in a reluctance for the heavier broilers to jump onto the pad. Note that a standard deviation between the measured local weight and the slaughter weight of more than 100g, equivalent of 5% of the total slaughter weight, is not uncommon.

The slaughter weight is considered very accurate and is included by overriding the last measured local weight at sample $k = N_{s,b}$ of each batch. Extra emphasis is then placed on the slaughter weight at sample $k = N_{s,b}$ in the cost function, while samples onwards of $N_\phi \in \mathbb{Z}_+$ are weighted less. A function achieving the desired behavior is given by

$$\phi(k) = \begin{cases} 1, & k < N_\phi \\ 1 + (N_{s,b} - N_\phi)(\gamma - 1), & k = N_{s,b} \\ \gamma, & \text{otherwise} \end{cases} \tag{7}$$

where $\phi : \mathbb{Z}_+ \to \mathbb{R}$ is the cost shaping function, $N_\phi \in \mathbb{Z}$ is the start cost shaping sample number, and $\gamma \in]0; 1[$ is the cost shaping weight. On Figure 3 an example of (7) is depicted. Note that $P_p \leq N_\phi$ for all p is required to avoid changing the relative weighting of other variables, such as feed consumption,

as

$$\frac{1}{N_{s,b} - P_p + 1} \sum_{k=P_p}^{N_{s,b}} \phi(k) = \frac{1}{N_{s,b} - P_p + 1} \sum_{k=P_p}^{N_{s,b}} 1 = 1.$$

Rewriting (6) to include the slaughter weight and the weight shaping function $\phi(k)$ given by (7), results in

$$\mathcal{W}^*(\mathcal{B}_t, N_\phi, \gamma) = \arg\min_{\tilde{\mathcal{W}}} \sum_{b \in \mathcal{B}_t \setminus \min\{\mathcal{B}_t\}} \frac{E_b^*(\tilde{\mathcal{W}}, N_\phi, \gamma)}{\#\mathcal{B}_t - 1} \tag{8}$$

with

$$E_b^*(\tilde{\mathcal{W}}, N_\phi, \gamma) = \sum_{p=1}^{N_P} \sum_{k=P_p}^{N_{s,b}} \frac{\mathcal{E}_b}{N_{r,b}} + \bar{\alpha} \|\tilde{\mathcal{W}}\|^2$$

$$\mathcal{E}_b = \sum_{i=1}^{N_y} \begin{cases} \|\Gamma_b - \hat{y}_i\|_2^2 \, \phi(k), & k = N_{s,b} \wedge i = i_w \\ \|y_i - \hat{y}_i\|_2^2 \, \phi(k), & k \neq N_{s,b} \wedge i = i_w \\ \|y_i - \hat{y}_i\|_2^2, & \text{otherwise} \end{cases}$$

where $y_i = y_i[k \mid b]$ is the target output, $\hat{y}_i = \hat{y}_i[k \mid P_p, b, \tilde{\mathcal{W}}]$ is the model output, Γ_b is the slaughter weight of batch b, and i_w is the index of the measured weight.

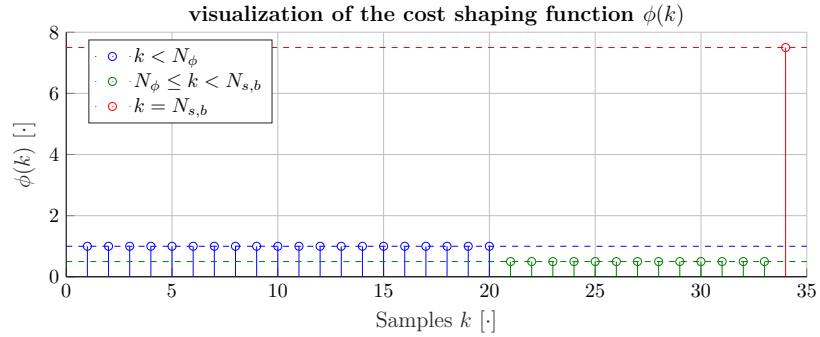


Figure 3: Visualization of the cost shaping function $\phi(k)$ with $N_\phi = 20$, $N_{s,b} = 34$ and $\gamma = 0.5$. The blue, green and red values correspond to a separate case of (7).

2.5 Ensemble Forecasting

To make the forecasting algorithm more robust against weight initialization effects, the ensemble mean of $N_m \in \mathbb{Z}_+$ sub-models are used to represent the "true" model output. Each sub-model is trained on the same training data, but with different initial weights. The ensemble forecasting model mean of trial

$t \in \mathbb{Z}_+$ and output $l \in \mathbb{Z}_+$ for the two algorithms are given by

$$\hat{\bar{y}}_l[k \mid t, p] = \sum_{i=1}^{N_m} \frac{\hat{y}_l[k \mid t, p, \mathcal{W}_i(\mathcal{B}_t)]}{N_m} \text{ and} \tag{9a}$$

$$\hat{\bar{y}}^*_l[k \mid t, p, N_\phi, \gamma] = \sum_{i=1}^{N_m} \frac{\hat{y}_l[k \mid t, p, \mathcal{W}^*_i(\mathcal{B}_t, N_\phi, \gamma)]}{N_m}, \tag{9b}$$

where $\hat{\bar{y}}_l[k \mid t, p]$ and $\hat{\bar{y}}^*_l[k \mid t, p, N_\phi, \gamma]$ are the ensemble forecasting mean of the nominal and improved algorithm, $\mathcal{W}_i(\mathcal{B}_t)$ is the i'th trained sub-model weight using the nominal algorithm (6), and $\mathcal{W}^*_i(\mathcal{B}_t, N_\phi, \gamma)$ is trained using the improved algorithm (8).

3 Experimental Results

3.1 Data Description

The data used in this work is gathered from an ≈20 year old state-of-the-art broiler house located in northern Denmark and consists of 35 batches corresponding to more than 4.5 years of production data. The data is from the same location as [Johansen et al., 2019], but it includes the measured slaughter weight in addition to 6 more batches. An overview of the data is depicted on Figure 4, where the inputs consist of inside and outside temperature, humidity, light intensity, and CO2, while the outputs consist of broiler weight, cumulative feed, and water consumption.

Output Data Description

The weight, cumulative feed and water consumption are all strongly positively correlated with each other, which makes intuitive sense given that feed and water are both required for broiler growth. The broiler weight standard deviation is increasing with time until day 31, where the mean weight drops a little – this is caused by the fastest growing batches being terminated as the broilers have reached their target weight and are therefore slaughtered.

On Figure 5, the last locally measured weight is compared with the measured slaughter weight. In this case, the local measured weight has been quite nicely corrected with the previously mentioned correction factor, as the mean error is quite low. However, a standard error of 115.9g is quite significant, which suggests that there is still room for improvements as the "behavior" constant has not corrected the biased weight problem completely.

Input Data Description

The fundamental assumption behind this work is that the inputs can explain the difference in the otherwise strongly correlated outputs.

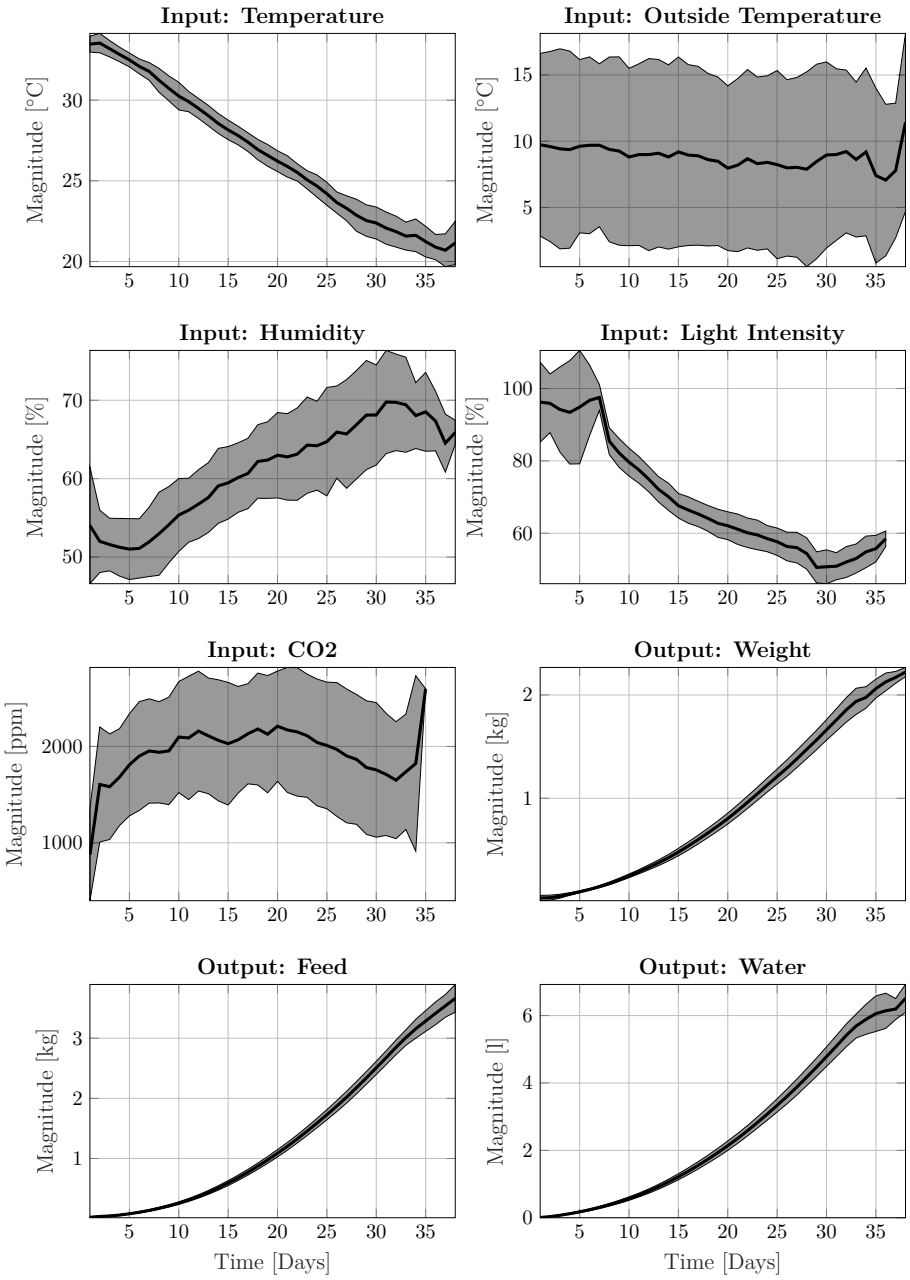


Figure 4: Overview of the 35 batches used for this forecasting study. The bold line denotes the mean value and the shaded area denotes a standard deviation of the available data. Note that inputs refer to U and outputs refer to Y in (1).

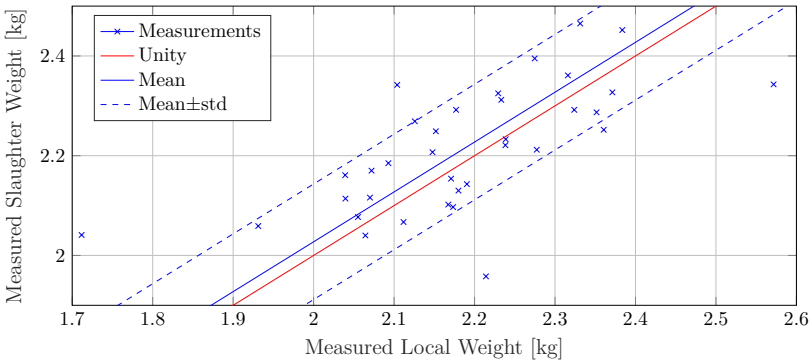


Figure 5: Comparison between measured local weight and slaughter weight, with mean difference of 27.4g with a standard deviation of 115.9g, and median error of 45g.

The inside temperature is known to be the most influential climate variable for broiler growth in the industry. The broiler supplier recommends an almost linearly decreasing temperature, starting at 30 °C at day 0 and ending at 20 °C at day 27, where it remains for the duration of the batch. This particular broiler house has a higher initial temperature of 34 °C with a standard deviation of around 1°C, and a terminal temperature of 20°C at day 34 instead of 27. Hence, this location uses higher than recommended temperature.

The outside temperature is known to influence the climate control systems' ability to regulate the inside climate conditions, as outside air is used for ventilation. When averaging the outside temperature, it is very close to normally distributed with mean ≈9 °C and standard deviation ≈5 °C.

The mean humidity increases almost linearly from 50% to 70% throughout the batch and is positively correlated with the amount of biological material in the house. As the temperature decreases, less ventilation is often possible, resulting in higher humidity.

The mean light intensity decreases from ≈100% to 50% throughout the batch with a quite narrow standard deviation.

The mean CO2 generally start out at around 1500 ppm at day 0, peaks at day 15-20 at around 2000 ppm, and ends at 1500 ppm. European law states that CO2 levels cannot exceed 3000 ppm, so the influence of this variable might be limited.

Note that compared to [Johansen et al., 2017] and [Johansen et al., 2019], both the heating rate and ventilation rate are not included in this work, as excluding them results in more accurate forecasts. This is caused by a mismatch between the future substituted inputs from the prediction batch and the actual future values, sometimes resulting in highly unrealistic scenarios.

3.2 Default Algorithm Configuration

The default configuration from [Johansen et al., 2019] is unaltered unless specified, with the aim of making the results somewhat comparable. It uses $N_B = 10$ training batches, $N_m = 64$ sub-models, $N_h = 7$ neurons in the hidden layer, and $N_P = 5$ sub-batch training start times at day $P = \{2, 7, 14, 21, 28\}$. The sample interval is changed from $T_s = 3$ hours in [Johansen et al., 2019] to $T_s = 12$ hours to reduce training time.

The input variable selection algorithm selects up to 2 input variables from the available inputs, e.g. temperature denoted by $u_i[k \mid b]$ with index i, and up to 2 lags are selected per input, e.g. $u_i[k-1 \mid b]$ and $u_i[k-3 \mid b]$. In [Johansen et al., 2019] 3 input variables were selected, which is reduced to 2 in this study as fewer input candidates are used.

3.3 Test Evaluation Criteria

The objective of this study is to investigate the proposed algorithm's slaughter weight forecasting ability. The evaluation metric is root mean square error (RMSE), as it can be interpreted as the standard error, similar to the training cost function. The forecasting RMSE of all N_t tests between the slaughter weight Γ_t at sample $N_{s,t}$ and the weight forecast at sample $N_{s,t}$ from the prior $\Delta \in \mathbb{Z}_+$ samples, i.e. $p \in \{N_{s,t} - \Delta, \ldots, N_{s,t} - 1\}$, is defined by

$$\hat{\bar{\mathcal{J}}}_\Delta = \sqrt{\sum_{t=1}^{N_t} \sum_{p=N_{s,t}-\Delta}^{N_{s,t}-1} \frac{\left(\hat{\bar{y}}_{i_w}[N_{s,t} \mid t, p] - \Gamma_t\right)^2}{N_t \, \Delta}} \tag{10a}$$

$$\hat{\bar{\mathcal{J}}}^*_\Delta = \sqrt{\sum_{t=1}^{N_t} \sum_{p=N_{s,t}-\Delta}^{N_{s,t}-1} \frac{\left(\hat{\bar{y}}^*_{i_w}[N_{s,t} \mid t, p, N_\phi, \gamma] - \Gamma_t\right)^2}{N_t \, \Delta}} \tag{10b}$$

where $\hat{\bar{\mathcal{J}}}_\Delta$ is the forecasting RMSE of the nominal algorithm and $\hat{\bar{\mathcal{J}}}^*_\Delta$ is the RMSE of the improved algorithm. Hence, it is possible to investigate forecasting performance at different forecasting horizons, e.g. a 7 day horizon with a sample interval T_s of 12 hours is obtained with $\Delta = 14$. Furthermore, the RMSE between the measured local weight and the slaughter weight is calculated by

$$\mathcal{J}_0 = \sqrt{\sum_{t=1}^{N_t} \frac{\left(y_{i_w}[N_{s,t} \mid t] - \Gamma_t\right)^2}{N_t}}, \tag{11}$$

which allows for comparison of the algorithms with the locally measured weight used for training prior to slaughter.

3.4 Test Description

In the following, several tests are specified with the intention of providing insights into effective configuration of the improved slaughter weight forecasting

algorithm. Each test explores the effect of one or two parameters by changing it from the default configuration described in Section 3.2.

Test #1 investigates the effect of including the slaughter weight through the weight shaping function as detailed in Section 2.4. The measured local weight is used as a baseline through $\mathcal{J}_1$ with $\Delta = 1$. The statistical significance of using the improved algorithm is determined using an ANOVA-factor analysis with 5% significance level of the residuals, i.e., $\hat{\tilde{y}}^{\star}_{w} - \Gamma$. This test assumes normally distributed variables, which are checked using the Anderson-Darling normal distribution test with a 5% significance level.

Test #2 investigates different settings of the weight shaping parameters N_ϕ and γ in (7). Setting $N_\phi = \text{day}\ \{12, 15, 18, 21\}$, the aim is to determine the bias onset – currently, it is believed to be around day 15-21. Setting $\gamma = \{0, 0.1, 0.2\}$, the aim is to investigate the influence of γ.

Test #3 investigates the effect of different number of training batches N_B, and subsequently the parameter-drift rate – as discussed in Section 2.3. $N_B = 10$ training batches were heuristically chosen in [Johansen et al., 2019] and [Johansen et al., 2017] – corresponding to ≈1.5 years of production data.

Test #4 investigates the number of hidden neurons N_h, which has been heuristically chosen to be $N_h = 7$ in [Johansen et al., 2019] and [Johansen et al., 2017].

Test #5 investigates sample interval T_s, which has been heuristically chosen to be 3 hours in [Johansen et al., 2019] and [Johansen et al., 2017].

Note that multiple test-iterations can be combined to form a "steepest descent"-like parameter optimization algorithm. Such an approach has only been carried out for the two new parameters N_ϕ and γ in the following. An exhaustive search is not practically feasible, as the test suite takes around 33 hours to complete.

3.5 Test Results

The test results of the five test cases described in Section 3.4 are depicted on Table 1.

Test #1

On Figure 6 a box plot of the slaughter weight prediction errors for Test #1 with $\Delta = 1$ is depicted. It shows that the median measured weight error is 50g with an interquartile range of 145g – suggesting that the measured weight has a large variation. The nominal algorithm has a median weight error of -154g and interquartile range of 100g – suggesting that the nominal algorithm is highly

Table 1: Experimental Test Results

The colored circles denote the relative test score, with green 🟢 being the best and red 🔴 the worst performing, while † denotes the default configuration. Note that Δ is the forecasting horizon in (10).

	Type	**Last Sample ($\Delta = 1$)**	**Last 7 Days ($\Delta = 14$)**
Test #1	$\mathcal{J}_0$	🟡 106.5	🟡 106.5
	$\hat{\bar{\mathcal{J}}}_\Delta$	🔴 162.4	🔴 167.1
	$\hat{\bar{\mathcal{J}}}^{\star\,\dagger}_\Delta$	🟢 65.4	🟢 73.6

	N_ϕ [Day]	γ [·]	$\hat{\bar{\mathcal{J}}}^{\star}_1$ [g]	$\hat{\bar{\mathcal{J}}}^{\star}_{14}$ [g]
Test #2	12	0	🟢 75.0	🟢 76.7
	12	0.1	🟠 90.0	🟡 89.6
	12	0.2	🟠 90.8	🟡 88.5
	15†	0†	🟢 65.4	🟢 73.6
	15	0.1	🟡 83.2	🟡 91.1
	15	0.2	🔴 93.4	🟡 90.8
	18	0	🟢 71.1	🟢 76.2
	18	0.1	🟡 85.4	🟡 87.3
	18	0.2	🟠 89.2	🟠 91.8
	21	0	🟢 77.6	🟢 83.6
	21	0.1	🟠 88.6	🔴 94.0
	21	0.2	🔴 95.6	🔴 96.6

	N_B [·]	$\hat{\bar{\mathcal{J}}}^{\star}_1$ [g]	$\hat{\bar{\mathcal{J}}}^{\star}_{14}$ [g]
Test #3	14	🟡 76.0	🟡 86.4
	12	🟢 66.2	🟢 69.7
	10†	🟢 65.4	🟢 73.6
	8	🟡 77.7	🟡 84.6
	6	🟢 65.3	🟢 77.0
	4	🔴 116.8	🔴 150.5

	N_h [·]	$\hat{\bar{\mathcal{J}}}^{\star}_1$ [g]	$\hat{\bar{\mathcal{J}}}^{\star}_{14}$ [g]
Test #4	11	🔴 73.0	🔴 79.1
	9	🟠 72.0	🔴 78.6
	7†	🟢 65.4	🟡 73.6
	5	🟢 66.2	🟢 72.2
	3	🟡 71.4	🟢 71.7

	T_s [Hours]	$\hat{\bar{\mathcal{J}}}^{\star}_1$ [g]
Test #5	24	🔴 90.7
	12†	🟢 65.4
	6	🟡 70.8
	3	🟢 65.0

negatively biased for slaughter weight forecasting. The modified algorithm has a median weight error of -8g and interquartile range of 96g – suggesting that using the slaughter weight greatly increases the accuracy of the algorithm but not necessarily the precision. Comparing the RMSE for the most recent sample ($\Delta = 1$) and last 7 days ($\Delta = 14$) listed in Table 1 shows that the improved algorithm is superior to both the measured local weight and the nominal algorithm.

The Anderson-Darling test cannot reject that any of the residuals with $\Delta = 1$ of the three groups for Test #1 are normally distributed at a 5% significance level. According to the ANOVA factor analysis, the means of all groups are significantly different at a 5% significance level. The means of the nominal and improved method is significantly different with a p-value of 0.044, which might be susceptible to change with growing sample sizes.

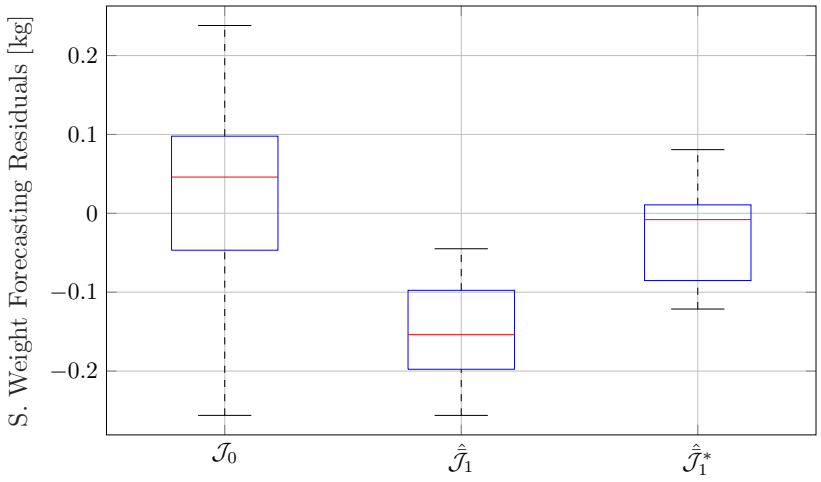


Figure 6: Box plot of Test #1 residuals of the measured weight $\mathcal{J}_0$, and both the nominal algorithm $\hat{\tilde{\mathcal{J}}}_1$ and improved algorithm $\hat{\tilde{\mathcal{J}}}_1^*$ with $\Delta = 1$. See Section 3.3 for details.

Test #2

The best performing weight shaping parameters are $N_\Phi = 15$ and $\gamma = 0$. The best performing configurations with fixed N_Φ uses $\gamma = 0$, which makes sense as it avoids discontinuous jumps between the slaughter weight and the measured weight during training. If the shape of $\phi(k)$ was changed to monotonically decrease from 1 to 0 between N_Φ and $N_{s,t}-1$, then better performance might be obtained. Similarly, the best performing configurations with fixed γ uses $N_\Phi = 15$, where a lower or higher value results in decreased performance. This coincides nicely with the previously stated heuristic claim that the measured broiler weight is biased onwards of day 15.

Test #3

The best choice of number of batches with $\Delta = 1$ and $\Delta = 14$ are $N_\Phi = \{6, 10, 12\}$. As $N_B = 4$ produces significantly worse results than $N_B = 6$, it is recommended to use more than 4 batches for training. Similarly, given that $N_B = 14$ and $N_B = 8$ produce worse results, a N_B around $\{10, 12\}$ is recommended because they consistently have good $\hat{\bar{\mathcal{J}}}_1^*$ and $\hat{\bar{\mathcal{J}}}_{14}^*$. However, as $N_B = 6$ has good performance, the valid range is expected to be between 6 and 12. This result agrees well with the heuristic parameter drift claim discussed in Section 3.4.

Test #4

The best choice of number of hidden neurons are $N_h = \{5, 7\}$, where $N_h = 5$ is better with a very short forecasting horizon and $N_h = 7$ is better for longer forecasting horizon. Due to a similar performance it could be advantageous to use $N_h \approx 5$, as it reduce the maximal number of model parameters in $\mathcal{W}$ from 248 to 178, before input selection is applied to reduce the input space.

Test #5

According to the test, the best choice of sampling interval are $T_s = \{3, 12\}$ hours. Given a huge training time difference between the two, a sampling interval of $T_s = 12$ hours is preferred. However, a sampling interval of $T_s = 24$ hours is not recommended.

4 Concluding Remarks

In this paper, a broiler weight forecasting algorithm, based on dynamic neural network, has been adapted for slaughter weight forecasting and tested on 4.5 years of production data from a single industrial scale *state-of-the-art* broiler farm. By emphasizing the slaughter weight significantly more than the locally measured weight during training, which is commonly known to be negatively biased onwards of day 15-21, it was possible to significantly reduce the forecasting root mean squared error from 162.4g to 65.4g. Different algorithm settings were investigated to provide insights into the optimal algorithm configuration, where we among others found that the measurement bias appears to start around day 15.

Future work includes using the model for growth optimization, disease detection using fault detection algorithms by including antibiotic administration information, along with automatic data-filtration algorithms to remove/identify corrupted/false production data.

References

J. M. Aerts, M. Lippens, G. De Groote, J. Buyse, E. Decuypere, E. Vranken, and D. Berckmans. "Recursive prediction of broiler growth response to feed intake by using a time-variant parameter estimation method". *Poultry Science*, 82:40–49, 2003. ISSN 0032-5791.

F. Burden and D. Winkler. "Bayesian regularization of neural networks". In *Methods in Molecular Biology*, pages 23–42. Springer Science + Business Media, 2008. doi:10.1007/978-1-60327-101-1_3.

A. Chedad, E. Vranken, J.-M. Aerts, and D. Berckmans. "Behaviour of chickens towards automatic weighing systems". *IFAC Proceedings Volumes*, 33(29):207 – 212, 2000. doi:https://doi.org/10.1016/S1474-6670(17)36778-2.

A. Chedad, J.-M. Aerts, E. Vranken, M. Lippens, J. Zoons, and D. Berckmans. "Do heavy broiler chickens visit automatic weighing systems less than lighter birds?". *British Poultry Science*, 44(5):663–668, dec 2003. doi:10.1080/00071660310001643633.

T. Demmers, Y. Cao, S. Gauss, J. Lowe, D. Parsons, and C. Wathes. "Neural predictive control of broiler chicken growth". *IFAC Proceedings Volumes*, 43(6):311–316, 2010. doi:10.3182/20100707-3-be-2012.0061.

J. Grahovac, A. Jokić, J. Dodić, D. Vučurović, and S. Dodić. "Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks". *Renewable Energy*, 85:953–958, jan 2016. doi:10.1016/j.renene.2015.07.054.

S. C. Hernandez, J. A. Bueno, E. N. Sanchez, and L. Diaz-Jimenez. "State estimation by artificial neural networks in a continuous bioreactor". *IFAC Proceedings Volumes*, 46(31):215–220, 2013. doi:10.3182/20131216-3-in-2044.00033.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Broiler weight forecasting using dynamic neural network models with input variable selection". *Journal of Computers and Electronics in Agriculture*, 2019. doi:10.1016/j.compag.2018.12.014.

V. V. Nair, H. Dhar, S. Kumar, A. K. Thalla, S. Mukherjee, and J. W. Wong. "Artificial neural network based modeling to evaluate methane yield from biogas in a laboratory-scale anaerobic bioreactor". *Bioresource Technology*, 217:90–99, oct 2016. doi:10.1016/j.biortech.2016.03.046.

R. Nasimi and R. Irani. "Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks". *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14):1604–1611, may 2014. doi:10.1080/15567036.2011.592903.

R. C. Newberry, J. R. Hunt, and E. E. Gardiner. "Behaviour of roaster chickens towards an automatic weighing perch". *British Poultry Science*, 26(2):229–237, apr 1985. doi:10.1080/00071668508416808.

D. Nguyen and B. Widrow. "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights". In *1990 IJCNN International Joint Conference on Neural Networks*. Institute of Electrical & Electronics Engineers (IEEE), 1990. doi:10.1109/ijcnn.1990.137819.

OECD. *OECD-FAO Agricultural Outlook 2018-2027*. OECD Publishing, jul 2018. doi:10.1787/agr_outlook-2018-en.

M. Sadeghassadi, C. J. Macnab, B. Gopaluni, and D. Westwick. "Application of neural networks for optimal-setpoint design and MPC control in biological wastewater treatment". *Computers & Chemical Engineering*, 115:150–160, jul 2018. doi:10.1016/j.compchemeng.2018.04.007.

K. Stacey, D. Parsons, A. Frost, C. Fisher, D. Filmer, and A. Fothergill. "An automatic growth and nutrition control system for broiler production". *Biosystems Engineering*, 89(3):363–371, nov 2004. doi:10.1016/j.biosystemseng.2004.07.006.

H. Wang, X. Yan, H. Chen, C. Chen, and M. Guo. "Chlorophyll-a predicting model based on dynamic neural network". *Applied Artificial Intelligence*, 29(10):962–978, nov 2015. doi:10.1080/08839514.2015.1097142.

Paper D

Broiler Growth Optimization using Optimal Iterative Learning Control

Simon V. Johansen[a,b], Jan D. Bendtsen[b] and Jesper Mogensen[b]

[a]SKOV A/S, Hedelund 4, Glyngøre, Denmark
[b]Department of Control and Automation, Aalborg University, Denmark

***Abstract*—*In this paper the first recorded attempt at optimizing broiler growth using iterative learning control under state-of-the-art production conditions is presented. The work is motivated by a significant predicted increase in global broiler meat, where existing optimization techniques are incompatible with state-of-the-art broiler production. The proposed method regulates broiler growth using broiler house temperature based on norm optimal iterative learning control, which is a model based control technique. To compensate for the lack of mathematical broiler growth models in scientific literature, dynamic neural network models are used, which is a data driven modeling technique. Practical results from a state-of-the-art broiler house appear promising, but not conclusive, although a maximum decrease in required feed of 2.5% was obtained.*

Keywords*—*Iterative learning control; Biological systems; Neural networks

Published in *Proceedings of American Control Conference*, 2019.

Contents

1 Introduction

Global poultry meat production is predicted to increase by 18% between 2015-17 and 2027 to a total of 139 billion kg [OECD, 2018, pp. 37], of which broiler meat represents the majority. This motivates the need for better broiler optimization techniques, where available research exclusively focuses on broiler weight control through feed uptake regulation. However, this research is not compatible with state-of-the-art broiler industry, as ad Libitum feeding regimes are primarily used, i.e., feed uptake regulation is not possible. Broiler climate regulation, e.g. temperature, humidity and CO_2, is compatible with state-of-the-art industry, where correct management is known to lower the amount of feed, water and electricity required to produce a mature broiler significantly – making such methods environmentally and financially desirable. This paper documents the first experimental results on optimizing broiler growth through temperature regulation in scientific literature.

Broiler production is a biological batch process, which is known to be highly nonlinear and time varying, making traditional linear modeling and control tools inadequate [Bonné et al., 2004]. Iterative Learning Control (ILC) is a high precision control tool that iteratively improves the performance of repetitively operating dynamic systems, which makes it a potential candidate for broiler growth optimization. It is a relatively recent but well-established field of study in control theory and can be traced back to [Uchiyama, 1978] and [Arimoto et al., 1984]. The objective for the batch process is to track a reference trajectory as closely as possible, which is known or experienced to yield good performance. ILC solves this problem by applying feedback from the error of the current and past batches [Wang et al., 2009]. In [Duran-Villalobos and Lennox, 2013] ILC has been applied successfully to a fermentation process. In [Maeda et al., 2015] an ILC algorithm is combined with a disturbance observer to remove near-repetitive disturbances, and it is tested and verified on an excavator. Other recent and noteworthy contributions include [Márquez-Vera et al., 2014], [Lim and Barton, 2014], and [Shen and Wang, 2014]. Model free ILC is not an option, as the input sign depends on the input, e.g., too hot or cold temperature decrease performance. We use norm optimal ILC, which is a popular branch of ILC, where the ILC input is optimal respect a cost function. The idea was formalized in [Furuta and Yamakita, 1986] and further developed in [Amann et al., 1996]. Noteworthy application examples include [Zhang, 2008], [van Zundert et al., 2016] and [Bolder and Oomen, 2015].

Norm optimal ILC is a model based technique, and since no mathematical models exist of the broiler growth process in scientific literature, ensemble dynamic neural network (DNN) class models are used as a viable alternative. Such techniques have successfully been applied to model complex biological processes and have been extensively studied in scientific literature – see [Johansen et al., 2019] and [Johansen et al., 2017] for detailed explanations and descriptions of such models. Combining ILC schemes with a data driven modeling method is not a novel idea – recent examples include [Bolder and Oomen,

2015] and [Xu et al., 2013]. A similar combination of optimal ILC and ensemble dynamic neural network model, as used in this work, has been proposed in [Zhang, 2008], but has not been applied to broiler production before.

In this work broiler growth is optimized using constrained norm optimal ILC by regulating temperature – arguably the most influential climate variable in state-of-the-art broiler production. The model used for ILC is a data driven dynamic neural network model with automatic input variable selection, which has been described in detail in [Johansen et al., 2019]. Experimental results of applying the proposed method to state-of-the-art industrial broiler production are obtained.

In Section 2 a data driven norm optimal ILC algorithm is described, which is adapted to broiler growth optimization in Section 3. Experimental results are described in Section 4, while concluding remarks are provided in Section 5.

2 Method

2.1 Iterative Learning Control (ILC)

In super-vector notation, consider the plant with the tracking error given by

$$Y_k(U_k) = P_k U_k + K_k \text{ and} \tag{1a}$$
$$E_k(U_k) = R - Y_k(U_k), \tag{1b}$$

with

$$Y_k = \begin{bmatrix} y_k[N_s]^T & \cdots & y_k[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_Y}, \tag{2a}$$
$$U_k = \begin{bmatrix} u_k[N_s]^T & \cdots & u_k[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_U} \text{ and} \tag{2b}$$
$$R = \begin{bmatrix} r[N_s]^T & \cdots & r[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_Y}, \tag{2c}$$

where $(\cdot)_k$ denotes the k'th trial, N_s is the start sample, and N_e is the end sample. $Y_k \in \mathbb{R}^{N_Y}$, $U_k \in \mathbb{R}^{N_U}$, $R \in \mathbb{R}^{N_Y}$ and E_k:$\mathbb{R}^{N_U} \to \mathbb{R}^{N_Y}$ are the super-vector output, input, reference and tracking error. $y_k[n] \in \mathbb{R}^{N_y}$, $u_k[n] \in \mathbb{R}^{N_u}$ and $r[n] \in \mathbb{R}^{N_y}$ are the n'th output, input and reference. $N_Y = N_y(N_e - N_s + 1)$ and $N_U = N_u(N_e - N_s + 1)$. $P_k \in \mathbb{R}^{N_Y \times N_U}$ is the super-vector system matrix, and $K_k \in \mathbb{R}^{N_Y}$ is effects unrelated to the input U_k.

ILC aims at iteratively finding the next input sequence U_{k+1} such that the output sequence Y_{k+1} converges to the desired reference sequence R in k, corresponding to

$$\lim_{k \to \infty} Y_{k+1}(U_{k+1}) = R. \tag{3}$$

In this work input constrained norm optimal ILC is used to calculate the next ILC input U_{k+1} by solving

$$\begin{aligned} U_{k+1} = \arg\min_{U} \; & \|E_k(U)\|^2_{W_E} + \|U - U_k\|^2_{W_{\Delta U}} \\ \text{s.t.} \quad & C_l \le U - U_k \le C_u \end{aligned} \tag{4}$$

where $\|x\|_Q = \sqrt{x^T Q x}$ is the weighted euclidean norm, $U \in \mathbb{R}^{N_U}$ is the optimization variable, $C_l \in \mathbb{R}^{N_U}$ and $C_u \in \mathbb{R}^{N_U}$ are the lower and upper bounds on the ILC input change relative to the last input U_k, $W_E \in \mathbb{R}^{N_Y \times N_Y}$ is the output error cost matrix, and $W_{\Delta U} \in \mathbb{R}^{N_U \times N_U}$ is the input change cost matrix.

In the following, (4) is formulated as a quadratic minimization problem. Rewriting the tracking error in terms of the input change, conveniently denoted $\Delta U = U - U_k$, results in

$$E_k(U) = R - P_k\left[U_k + \Delta U\right] - K_k = E_k(U_k) - P_k \Delta U.$$

Grouping the cost function in (4) by ΔU and $E_k(U_k)$ yields

$$\|\Delta U\|^2_{\left(P_k^T W_E P_k + W_{\Delta U}\right)} + \|E_k(U_k)\|^2_{W_E} - 2\, E_k(U_k)^T W_E P_k \Delta U,$$

which can be reformulated as a quadratic minimization problem with constraints by

$$\begin{aligned} U_{k+1} = U_k + \underset{\Delta U}{\arg\min}\ & \tfrac{1}{2}\|\Delta U\|^2_{Q_1} + Q_2^T \Delta U \\ \text{s.t.}\quad & C_l \le \Delta U \le C_u \end{aligned} \tag{5}$$

with

$$\begin{aligned} Q_1 &= 2P_k^T W_E P_k + 2W_{\Delta U} \text{ and} \\ Q_2 &= -2P_k^T W_E E_k(U_k). \end{aligned}$$

This problem can be solved by standard quadratic programming algorithms, such as Matlab's `quadprog` implementation. To get an idea of the solution of (5) without constraints, we solve for the norm optimal ILC input change ΔU in

$$0 = \frac{d\left(\frac{1}{2}\|\Delta U\|^2_{Q_1} + Q_2^T \Delta U\right)}{d\Delta U} = Q_1 \Delta U + Q_2,$$

resulting in the optimal solution given by

$$\Delta U = -Q_1^{-1} Q_2 = \overbrace{\left(P_k^T W_e P_k + W_{\Delta U}\right)^{-1} P_k^T W_E}^{L_k} E_k(U_k),$$

where $0 < P_k^T W_E P_k + W_{\Delta U}$ of which we impose $0 < W_{\Delta U}$ and $0 < W_E$. Note that apart from the definiteness, no particular structure is imposed on W_E and $W_{\Delta U}$, hence, a different weight can be assigned to each sample.

The unconstrained solution of (5) has the shape of

$$U_{k+1} = U_k + L_k E_k(U_k), \tag{6}$$

and is known as an first order ILC feedback law. It is illustrated on Figure 1. According to [Xu, 2011], this relatively simple first order control law is adequate, even for highly nonlinear and uncertain systems. Assuming static P, K and L, then

$$\begin{aligned} E_{k+1}(U_{k+1}) &= R - Y_{k+1}(U_{k+1}) \\ &= R - P\,U_{k+1} - K \\ &= R - P\,U_k - K - P\,L\,E_k(U_k) \\ &= (I - PL)\,E_k(U_k). \end{aligned} \tag{7}$$

Hence, to design a asymptotically stable ILC controller in the trial domain k, the eigenvalues of the matrix $(I - PL)$ must be inside the unit circle. The fastest converging controller is obtained with $L = P^{-1}$ [sung Ahn et al., 2006], which in the case of norm optimal ILC is obtained with $W_e = I$ and $W_{\Delta U} = 0$. This makes intuitive sense as $E_{k+1}(U_k) = \left(I - PP^{-1}\right)E_k(U_k) = 0$, i.e., it converges in a single trial. Practical issues often arise when calculating P^{-1}, as P is often ill conditioned, large and non-invertible, which norm optimal ILC does not suffer from to the same extend due to the regularization effect from the weighting matrices W_E and $W_{\Delta U}$.

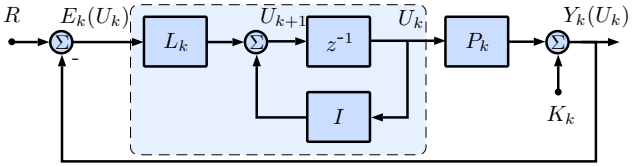


Figure 1: A graphical illustration of (1) and (6), where the ILC governed by (6) is highlighted for clarity.

2.2 Data Driven Model

The system matrix P_k required for ILC in (5) is unknown for broiler production, as no mathematical models exists in scientific literature, which is remedied by approximating P_k by the ensemble data driven model system matrix $\hat{\hat{P}}_k$ according to $P_k \approx \hat{\hat{P}}_k$. $N_b \in \mathbb{Z}_+$ prior batches are used for the data driven model generation as depicted on Figure 2. To calculate the next input U_{k+1} the most recent batch, k, is used to calculate $\hat{\hat{P}}_k$ using the model trained with batch $k{-}N_b{-}1, \ldots, k{-}1$.

Dynamic neural Network Model

The model used in this work is a dynamic neural network (DNN) and has been described in detail in [Johansen et al., 2019]. We only elaborate on its structure

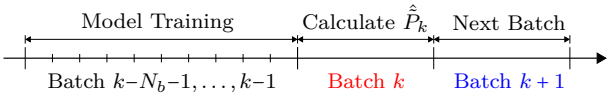


Figure 2: Visual segmentation of how batches are selected and used.

in the following and refer the reader to [Johansen et al., 2019] for the training specifics.

The particular type of DNN model used in this work can be classified as a discrete-time nonlinear ARMAX model of the form

$$\hat{y}_k[n+1] = \mathcal{N}(\hat{\mathcal{Y}}_k[n], \mathcal{U}_k[n], \mathcal{D}_k[n]) \qquad \forall N_s \leq n \tag{8a}$$

$$\hat{y}_k[n+1] = y_k[N_s] \qquad \forall N_s > n \tag{8b}$$

with

$$\hat{\mathcal{Y}}_k[n] = \begin{bmatrix} \hat{y}_k[n-\bar{m}_1]^T & \cdots & \hat{y}_k[n-\bar{m}_{N_{\bar{m}}}]^T \end{bmatrix}^T, \tag{9a}$$

$$\mathcal{U}_k[n] = \begin{bmatrix} u_k[n-\bar{n}_1]^T & \cdots & u_k[n-\bar{n}_{N_{\bar{n}}}]^T \end{bmatrix}^T \text{ and} \tag{9b}$$

$$\mathcal{D}_k[n] = \begin{bmatrix} d_k[n-\bar{n}_1]^T & \cdots & d_k[n-\bar{n}_{N_{\bar{n}}}]^T \end{bmatrix}^T, \tag{9c}$$

where $\mathcal{N}: \mathbb{R}^{N_y N_{\bar{m}}} \times \mathbb{R}^{N_u N_{\bar{n}}} \times \mathbb{R}^{N_d N_{\bar{n}}} \to \mathbb{R}^{N_y}$ is the DNN neural network model, $\mathcal{U}_k[n]$ and $\mathcal{D}_k[n]$ are delayed values of the input vector $u_k[n] \in \mathbb{R}^{N_u}$ and disturbance vector $d_k[n] \in \mathbb{R}^{N_d}$ corresponding to the $N_{\bar{n}}$ elements of $\bar{n} = \{\bar{n}_1, \cdots, \bar{n}_{N_{\bar{n}}}\} \in \mathbb{Z}_+$, $\hat{\mathcal{Y}}_k[n]$ is delayed values of the previous model output vector $\hat{y}_k[n] \in \mathbb{R}^{N_y}$ corresponding to the $N_{\bar{m}}$ elements of $\bar{m} = \{\bar{m}_1, \cdots, \bar{m}_{N_{\bar{m}}}\} \in \mathbb{Z}_+$, and N_s is the start sample.

The DNN model $\mathcal{N}$ is selected with one hidden layer with hyperbolic tangent activation function in the hidden layer and linear activation function in the output layer. In matrix-vector representation, (8) is written explicitly as

$$\hat{y}_k[n+1] = W^{\mathrm{o}} \tanh\big(\mathcal{X} + \theta^{\mathrm{h}}\big) + \theta^{\mathrm{o}} \text{ with} \tag{10}$$

$$\mathcal{X} = \sum_{i=1}^{N_{\bar{m}}} W^{\mathrm{h}}_{y,i} \hat{y}_k[n-\bar{m}_i] + \sum_{j=1}^{N_{\bar{n}}} W^{\mathrm{h}}_{ud,j} \begin{bmatrix} u_k[n-\bar{n}_j] \\ d_k[n-\bar{n}_j] \end{bmatrix},$$

where N_h is the number of neurons in the hidden layer, $\mathcal{X} \in \mathbb{R}^{N_h}$, $W^{\mathrm{o}} \in \mathbb{R}^{N_y \times N_h}$ is the output weights, $W^{\mathrm{h}}_{y,i} \in \mathbb{R}^{N_h \times N_y}$ is the delayed output weights, $W^{\mathrm{h}}_{ud,j} \in \mathbb{R}^{N_h \times (N_u + N_d)}$ is the delayed input weights, $\theta^{\mathrm{h}} \in \mathbb{R}^{N_h}$ is the hidden layer bias and $\theta^{\mathrm{o}} \in \mathbb{R}^{N_y}$ is the output bias.

The structure of $\mathcal{N}$ is automatically configured using mutual information based input variable selection, where $W^{\mathrm{h}}_{ud,j}$ and $W^{\mathrm{h}}_{y,i}$ can be sparsely populated to accommodate specific input and output lags, along with $\bar{n}$ and $\bar{m}$. The matrices and vectors $W^{\mathrm{h}}_{ud,j}$, W^{o}, $W^{\mathrm{h}}_{y,i}$, θ^{o} and θ^{h} are determined through training. For detailed information regarding the training and automatic configuration we redirect the reader to [Johansen et al., 2019].

DNN Super-vector Model

We linearize (8) in U_k to calculate $\hat{P}_k \approx P_k$ such that it can be used by the ILC algorithm described in Section 2.1. Applying a first order Taylor expansion in $\hat{U}_k$ yields

$$\hat{Y}_k(U) \approx \hat{Y}_k(U_k) + \hat{P}_k\left[U - U_k\right] = \hat{P}_k U + K_k \tag{11}$$

with

$$\hat{P}_k = \left.\frac{d\hat{Y}_k(U_k)}{dU_k^T}\right|_{U_k} \text{ and } K_k = \hat{Y}_k(U_k) - \hat{P}_k U_k,$$

where $\hat{Y}_k(U_k)$ is the simulation output of (8).

The DNN model described in Section 2.2 can be generalized as a nonlinear dynamic system by

$$x_k[n+1] = F(x_k[n],\, u_k[n],\, d_k[n]) \text{ and} \tag{12a}$$
$$\hat{y}_k[n] = H(x_k[n]) \tag{12b}$$

with the state vector $x_k[n]$ given by

$$\begin{aligned}
x_k[n] &= \left[x_{k,\hat{y}}[n]^T \quad x_{k,u}[n]^T \quad x_{k,d}[n]^T\right]^T \in \mathbb{R}^{N_x},\\
x_{k,\hat{y}}[n] &= \left[\hat{y}_k[n]^T \quad \cdots \quad \hat{y}_k[n-\max\{\bar{m}\}]^T\right]^T,\\
x_{k,u}[n] &= \left[u_k[n-1]^T \quad \cdots \quad u_k[n-\max\{\bar{n}\}]^T\right]^T \text{ and}\\
x_{k,d}[n] &= \left[d_k[n-1]^T \quad \cdots \quad d_k[n-\max\{\bar{n}\}]^T\right]^T,
\end{aligned}$$

where n is the sample number, $F{:}\,\mathbb{R}^{N_x}\times\mathbb{R}^{N_u}\times\mathbb{R}^{N_d} \to \mathbb{R}^{N_x}$ is the state transition equation, $H{:}\,\mathbb{R}^{N_x} \to \mathbb{R}^{N_y}$ is the measurement equation. Under these conditions $\hat{P}_k$ can be calculated through

$$\hat{P}_k = \begin{bmatrix}
0 & 0 & \cdots & 0\\
C_k[N_s{+}1]B_k[N_s] & 0 & \ddots & \vdots\\
\vdots & \ddots & \ddots & 0\\
C_k[N_e]\left[\prod_{i=N_e-1}^{N_s+1} A_k[i]\right]B_k[N_s] & \cdots & C_k[N_e{-}1]B_k[N_s] & 0
\end{bmatrix}$$

with:

$$A_k[n] = \frac{dx_k[n+1]}{dx_k^T[n]} \quad B_k[n] = \frac{dx_k^T[n+1]}{du_k[n]} \quad C_k[n] = \frac{dy_k[n]}{dx_k^T[n]} \tag{13}$$

In this work all model states are measured, which is exploited by constructing $x_{k,\hat{y}}[n]$ from the measured output $y_k[n]$, by assuming

$$\begin{aligned}
x_{k,\hat{y}}[n]^T &= \left[\hat{y}_k[n]^T \quad \cdots \quad \hat{y}_k[n-\max\{\bar{m}\}]^T\right]^T\\
&\approx \left[y_k[n]^T \quad \cdots \quad y_k[n-\max\{\bar{m}\}]^T\right]^T
\end{aligned} \tag{14}$$

when calculating $A_k[n]$, $B_k[n]$ and $C_k[n]$ – essentially linearizing along the measured trajectory. This might introduce unwanted noise and bias to the states used to calculate $\hat{P}_k$ and $\hat{K}_k$, but for this initial study it is preferred over having a long simulation horizon.

Ensemble Super-vector Model

We generate N_{models} models with different initial weights to reduce the risk of using a poor model, as the training of the model is not convex. For convenience, the super-vector notation of (11) for the i'th model at trial k is given by

$$\hat{Y}_{k,i}(U) = \hat{P}_{k,i}\, U + \hat{K}_{k,i}, \tag{15}$$

then the ensemble version of (11), denoted $\hat{\hat{Y}}_k$, equals

$$\hat{\hat{Y}}_k(U) = \frac{1}{N_{\text{models}}} \sum_{i=1}^{N_{\text{models}}} \hat{Y}_{k,i}(U) = \hat{\hat{P}}_k U + \hat{\hat{K}}_k \text{ with} \tag{16}$$

$$\hat{\hat{P}}_k = \frac{1}{N_{\text{models}}} \sum_{i=1}^{N_{\text{models}}} \hat{P}_{k,i} \text{ and } \hat{K}_k = \frac{1}{N_{\text{models}}} \sum_{i=1}^{N_{\text{models}}} \hat{K}_{k,i}.$$

This approximate ensemble model is used when calculating the next trial input U_{k+1} in (5), by setting $P_k \leftarrow \hat{\hat{P}}_k$, under the assumption that $\hat{\hat{P}}_k \approx P_k$.

3 Broiler Growth Optimization

Our aim is to optimize broiler growth by minimizing the feed conversion ratio (FCR) given by

$$\text{FCR} = \frac{\text{Feed Consumed}}{\text{Broiler Weight}}, \tag{17}$$

using the ILC algorithm presented in (5), which translates into using less feed to produce the same amount of broiler meat. This requires a suitable target reference R to be specified, which requires application specific knowledge about broiler growth and feed uptake behavior. To benchmark the broiler strains growth potential the manufacturer conducts small scale experiments under ideal climate conditions, which does not always transfer well to industrial scale manufacturing – especially considering that apparently identical broiler houses consistently produce different results.

The growth of the ROSS 308 broiler strain used in this study is described by a third order polynomial by the manufacturer. Our strategy is to fit a third order polynomial to feed consumption and broiler weight of the output data Y_k denoted Y_k^{fit}, which is "compressed" to emulate increased growth and "stretched" to emulate decreased feed uptake. The benefit of this approach is

that we try to modify the existing broiler behavior without forcing a "standard" reference, which might be unrealistic.

In this work Broiler Weight is maximized by compressing it 1% and feed consumption is minimized by stretching it 1%. We generate this reference at the first trial instance only to investigate convergence properties of the ILC algorithm, resulting in the reference at sample n given by

$$R[n] = \begin{bmatrix} Y^{\text{fit}}_{0,\text{ weight}}[n \cdot 1.01] \\ Y^{\text{fit}}_{0,\text{ feed}}[n \cdot 0.99] \end{bmatrix} \tag{18}$$

where $Y^{\text{fit}}_{0,\text{ weight}}$ and $Y^{\text{fit}}_{0,\text{ feed}}$ is the third order weight and feed consumption polynomial.

3.1 Input Constraints

In this section we describe how we formulate the input constraints C_l and C_u used in (5). The intended function of this feature is to limit the amount of input change in case of a poor model. Given that broilers are more sensitive to temperature changes early in the batch we use

$$c_u[n] = c_s + \frac{c_e - c_s}{N_e - N_s} \cdot (n - N_s) \text{ and} \tag{19a}$$

$$c_l[n] = -c_u[n], \tag{19b}$$

where $c_u[n]$ is an upper bound at sample n, $c_l[n]$ is a lower bound at sample n, c_s is the upper bound at sample N_s, $0 < c_s$ is the constraint at N_s while $0 < c_e$ is the constraint at N_e.

The input constraints for the temperature is conservatively set to N_s = day 0, N_e = day 34, c_s = 0.5 °C and c_e = 2 °C. An example of this is depicted on Figure 3.

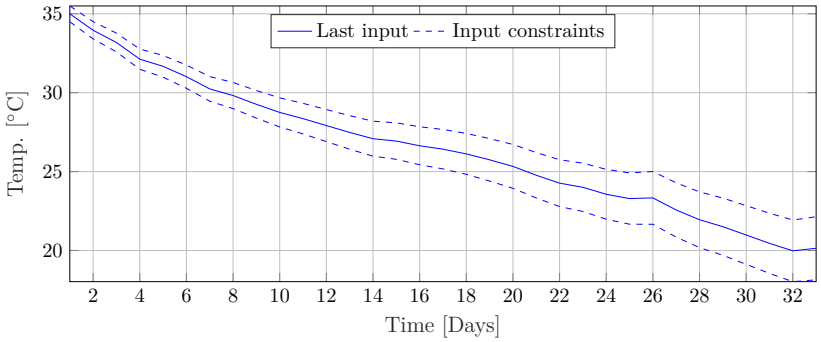


Figure 3: Example of temperature constraints with N_s = day 0, c_s = 0.5 °C, N_e = day 34 and c_e = 2 °C. The last input corresponds to $U_k[n]$, and the input constraints correspond to $U_k[n] + c_u[n]$ and $U_k[n] + c_l[n]$.

3.2 Weight Matrices

The norm optimal ILC weight matrices given by $W_{\Delta U}$ and W_E are selected to be constant diagonal matrices for simplicity using Bryson's rule according to

$$W_{\Delta U} = \operatorname{diag}(W_{\Delta U}[N_s], \cdots, W_{\Delta U}[N_e]) \text{ and} \tag{20a}$$
$$W_E = \operatorname{diag}(W_E[N_s], \cdots, W_E[N_e]) \tag{20b}$$

with

$$W_{\Delta U}[n] = (1\,^\circ\mathrm{C})^{-2} \text{ and}$$
$$W_E[n] = \operatorname{diag}\Big((12.5\,\mathrm{g})^{-2}, (1.6 \cdot 12.5\,\mathrm{g})^{-2}\Big),$$

where the constant 1.6 is a fairly common feed conversion ratio and the 12.5 g weight cost is heuristically chosen.

3.3 Limitations to the test-setup

The climate control system used in this test only has capacity for 8 linearly interpolated points for each reference, outside of the interval of these 8 points the value is equal to the closest point. As each point can be placed freely, we have good experience with using a knot-free B-spline algorithm[Luong] configured to use second order b-splines (linear interpolation) to find the 8 best fitting points. Note that the state-of-the-art broiler industry unanimously uses smooth temperature references, hence, this is not expected to pose any significant limitation.

It is a long standing problem for the broiler industry to measure broiler weight accurately onwards of day 15-21, where the broiler weight is negatively biased. It is typically corrected by multiplying a time dependent "behavior" constant to the measured weight, which naturally leads to some uncertainty. For this reason, the slaughter weight reported by the slaughter house is used to evaluate the performance of the algorithm, as it is an accurate sample mean produced by weighing all broilers from the batch. Note that the amount of feed is known to be accurately measured. Because the batches have different durations and slaughter weight, we normalize them by calculating the FCR at day 34 using

$$\mathrm{FCR@34}(y_f, y_w, t) = \frac{(y_f - k_f)\,34 \text{ days} + k_f t}{(y_w - k_w)\,34 \text{ days} + k_w t}, \tag{21}$$

where $y_f \in \mathbb{R}_+$ is the average feed consumed per broiler, $y_w \in \mathbb{R}_+$ is the average slaughter weight, t is the slaughter age in days, and $k_w = -1.110$ kg and $k_f = -3.081$ kg are correction factors. This expression is created using official regression formulas used by the Danish broiler industry[Det Danske Fjerkæraad, 2013, pp. 85].

3.4 Algorithm Configuration

A total of $N_{\text{models}} = 64$ DNN models are used in the model generation for the ILC design, which is configured as follows. The output variables, $\hat{y}_k[n]$, used equal: weight, cumulative feed consumption, and water consumption – water consumption is not used by the ILC algorithm. The input variable, $u_k[n]$, equals: (indoor) temperature. The disturbance variables, $d_k[n]$, is determined through variable selection from: humidity, outside temperature, heating rate, light intensity, CO_2, day number, and ventilation rate. The algorithm uses a sample interval of 24 hours. The input variable selection algorithm is configured to find up to 2 delays among the set $\{0, 1, 2\}$. Furthermore, the up to 3 significant disturbance variables are selected from the disturbance candidate list.

3.5 Algorithm Summary

Starting with $k = 1$, the Broiler ILC algorithm to calculate the next input U_{k+1} from the measured output $Y_k(U_k)$ is as follows:

1. Calculate the target reference R as described in Section 3.
2. Train N_{models} models according to Section 2.2.
3. Calculate the ensemble data driven system matrix $\hat{\bar{P}}_k$ in (16) as described in Section 2.2 and Section 2.2.
4. Calculate the next input U_{k+1} using the algorithm in (5), the constraints in Section 3.1, the target reference calculated in 1) and the weights as described in Section 3.2.
5. Convert U_{k+1} to 8 linearly interpolated points as described in Section 3.3
6. Get approval from a broiler application export and the farmer, then apply it to batch $k + 1$.
7. Record slaughter weight on completion of batch $k + 1$ then $k \leftarrow k + 1$ and goto step 2)

4 Experimental Results

A total of 4 trials where carried out on a state-of-the-art broiler house located in northern Denmark between September 2017 and March 2018, corresponding to almost 6 months of testing. The 10 most recent batches prior to the test are denoted pretest and are used for comparison.

The results from the house is depicted on Figure 4. Due to extraordinarily cold weather the heating system could not cope sufficiently in batch $k \in \{3, 4\}$. The measured temperature is depicted on the bottom left of Figure 4, where

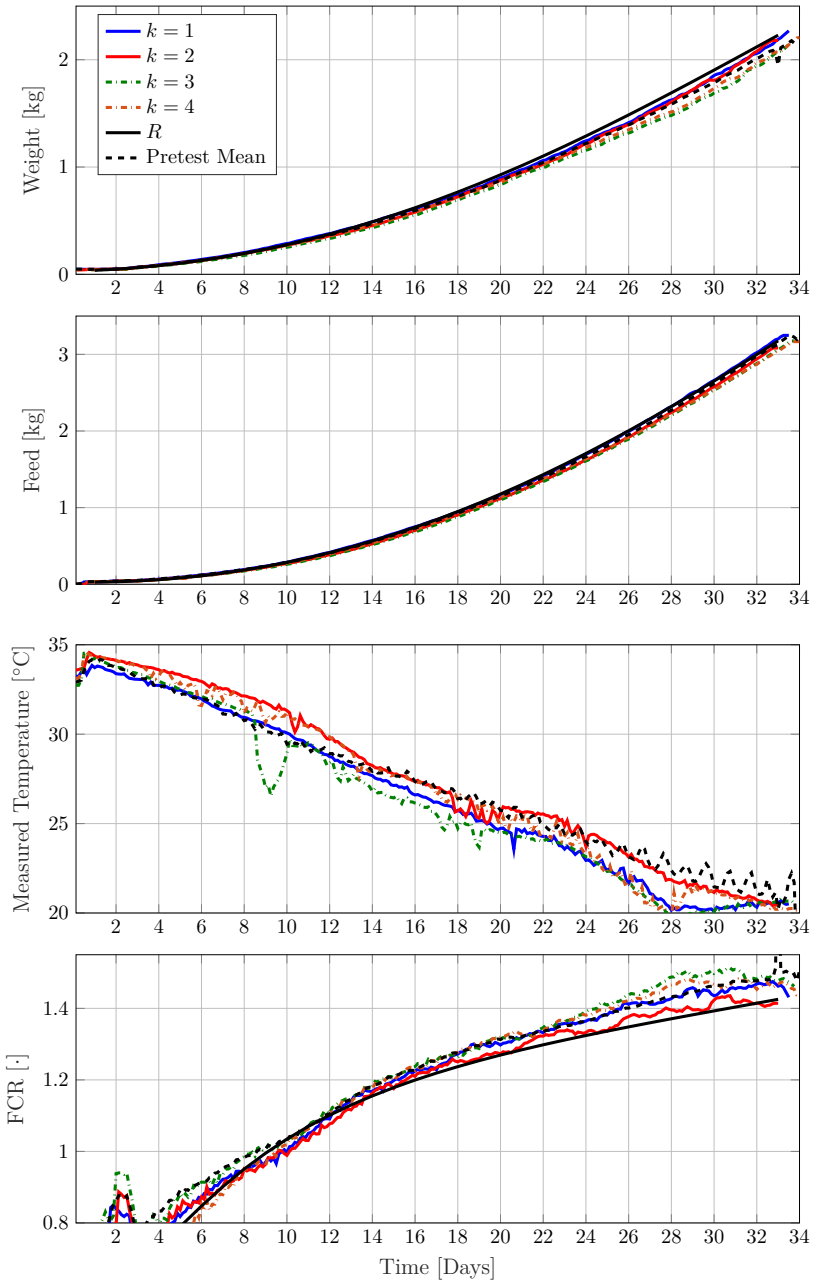


Figure 4: Results of applying the proposed ILC algorithm. The reference R is generated as described in Section 3.

batch $k = 3$ has a distinct negative spike at day 10 of 5 °C, and batch $k = 4$ has a small spike of 3 °C at day 4 and an offset of 2-3 $°C$ from the reference onwards of day 20. A broiler application expert considers this difference significant, especially in the beginning of the batch where the broilers are most sensitive to temperature deviations, hence, these batches are discarded when evaluating the proposed algorithms performance.

The test results are depicted on Figure 4. First off, trial $k \in \{1, 2\}$ progressively converge towards the reference R, as intended, where $k = 2$ is very close. This convergence behavior is unrealistic, as it is expected to asymptotically converge to R – this is, however, caused by a positive weight measurement bias as discussed previously in Section 3.3. The FCR for $k \in \{1, 3, 4\}$ are all quite close to the pretest mean.

The "stretching"-based reference generating strategy appears to be applied on a particularly high performing batch, which means that the ILC algorithm favors a heavier broiler with more feed compared to the pretest mean. This is not considered an issue, however, it is unexpected that FCR is lowered by increasing the feed uptake. The measured broiler weight is high for $k \in \{1, 2\}$, while the feed consumption only increases for $k = 1$.

In Table 1 the normalized test results are depicted. The FCR@34 is slightly worse for batch $k = 1$ with an FCR@34 increase of 0.013, which is within a standard deviation of the pretest mean. Batch $k = 2$ has a 0.025 lower FCR@34 compared to the pretest mean, corresponding to almost 2 standard deviations, which appear very promising. Hence, the trend appears to be negative from the first two batches, exactly as desired, but further testing is required to evaluate the performance of the proposed method.

	FCR@34 [·]	**Note**
Pretest	1.486 ± 0.017	mean ± std
R	1.445 (-0.043)	
$k = 1$	1.499 (0.013)	
$k = 2$	1.461 (-0.025)	
$k = 3$	1.544 (0.058)	Discarded
$k = 4$	1.549 (0.063)	Discarded

Table 1: Normalized Experimental Results. The pretest batches consists of 10 prior batches before the test, where (·) denotes the relative value to the pretest mean.

Finally, we would like to comment on the influence of the weight bias. The difference between the last measured weight in the broiler house and the slaughter weight for both the pretest and testing batches have a standard deviation of 76g. This is a known issue within the broiler industry, as discussed in Section 3.3, which we suspect makes it harder for the data driven algorithm to optimize FCR. For reference, a slaughter weight change of 76g for the pretest batches results in a change of FCR@34kg of around 0.05, which is alot compared to a standard deviation of 0.017 for the pretest batches.

5 Concluding Remarks

In this paper a data driven norm optimal ILC algorithm has been formulated with the aim of optimizing broiler FCR under state-of-the-art production conditions. Two trials have successfully been carried out, along with two discarded trials, on a state-of-the-art broiler house. The practical test results show a promising trend, but further testing is required to determine the effectiveness of the proposed method. Hence, the proposed algorithm's ability to optimize broiler FCR given more trials are not demonstrated conclusively yet.

Future work includes compensating for biased weight measurements and improving the performance of the data driven model, developing a broiler growth simulation framework to investigate the theoretical convergence properties of the proposed algorithm, and finally more practical testing to verify the algorithm's potential.

References

N. Amann, D. H. Owens, and E. Rogers. "Iterative learning control using optimal feedback and feedforward actions". *International Journal of Control*, 65(2):277–293, sep 1996. doi:10.1080/00207179608921697.

S. Arimoto, S. Kawamura, and F. Miyazaki. "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems". In *The 23rd IEEE Conference on Decision and Control*. IEEE, dec 1984. doi:10.1109/cdc.1984.272176.

J. Bolder and T. Oomen. "Data-driven optimal ILC for multivariable systems: Removing the need for l and q filter design". In *2015 American Control Conference (ACC)*. IEEE, jul 2015. doi:10.1109/acc.2015.7171880.

D. Bonné, B. Sten, and S. Jørgensen. "Data-driven modeling of batch processes". *IFAC Proceedings Volumes*, 37(1):589–594, 01 2004.

Det Danske Fjerkræraad. *Årsberetning 2013*. 2013. `https://danskfjerkrae.dk/om-fjerkraebranchen/det-danske-fjaerkraeraad/aarsberetning/aarsberetning-2013`.

C. A. Duran-Villalobos and B. Lennox. "Iterative learning modelling and control of batch fermentation processes". *IFAC Proceedings Volumes*, 46(32):511–516, dec 2013. doi:10.3182/20131218-3-in-2045.00010.

K. Furuta and M. Yamakita. "A design of learning control system for linear multivariable systems". *Transactions of the Society of Instrument and Control Engineers*, 22(12):1248–1255, 1986. doi:10.9746/sicetr1965.22.1248.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. “Broiler weight forecasting using dynamic neural network models with input variable selection”. *Journal of Computers and Electronics in Agriculture*, 2019. doi:10.1016/j.compag.2018.12.014.

I. Lim and K. L. Barton. “Pareto iterative learning control: Optimized control for multiple performance objectives”. *Control Engineering Practice*, 26:125–135, may 2014. doi:10.1016/j.conengprac.2014.01.011.

B. Luong. “Free-knot spline approximation”. MATLAB Central File Exchange. Retrieved 29 June 2017, `https://se.mathworks.com/matlabcentral/fileexchange/25872-free-knot-spline-approximation`.

G. J. Maeda, I. R. Manchester, and D. C. Rye. “Combined ILC and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation”. *IEEE Transactions on Control Systems Technology*, 23(5):1754–1769, sep 2015. doi:10.1109/tcst.2014.2382579.

M. A. Márquez-Vera, L. E. Ramos-Velasco, J. Suárez-Cansino, and C. A. Márquez-Vera. “Fuzzy iterative learning control applied in a biological reactor using a reduced number of measures”. *Applied Mathematics and Computation*, 246:608–618, nov 2014. doi:10.1016/j.amc.2014.08.072.

OECD. *OECD-FAO Agricultural Outlook 2018-2027*. OECD Publishing, jul 2018. doi:10.1787/agr_outlook-2018-en.

D. Shen and Y. Wang. “Survey on stochastic iterative learning control”. *Journal of Process Control*, 24(12):64–77, dec 2014. doi:10.1016/j.jprocont.2014.04.013.

H. sung Ahn, K. Moore, and Y. Chen. “LMI approach to iterative learning control design”. jul 2006. doi:10.1109/smcals.2006.250694.

M. Uchiyama. “Formulation of high-speed motion pattern of mechanical arm by trial”. *Trans. Soc. Instrum. Control Eng.*, 14(6):706–712, 1978. URL: `http://ci.nii.ac.jp/naid/10026860506/en/`.

J. van Zundert, J. Bolder, S. Koekebakker, and T. Oomen. “Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer”. *Mechatronics*, 38:76–90, sep 2016. doi:10.1016/j.mechatronics.2016.07.001.

Y. Wang, F. Gao, and F. J. Doyle. “Survey on iterative learning control, repetitive control, and run-to-run control”. *Journal of Process Control*, 19(10):1589–1600, dec 2009. doi:10.1016/j.jprocont.2009.09.006.

J.-X. Xu. “A survey on iterative learning control for nonlinear systems”. *International Journal of Control*, 84(7):1275–1294, jul 2011. doi:10.1080/00207179.2011.574236.

Z. Xu, J. Zhao, Y. Yang, Z. Shao, and F. Gao. “Optimal iterative learning control based on a time-parametrized linear time-varying model for batch processes”. *Industrial & Engineering Chemistry Research*, 52(18):6182–6192, may 2013. doi:10.1021/ie302561t.

J. Zhang. “Batch-to-batch optimal control of a batch polymerisation process based on stacked neural network models”. *Chemical Engineering Science*, 63(5):1273–1281, mar 2008. doi:10.1016/j.ces.2007.07.047.

Paper E

Broiler Growth Optimization using Norm Optimal Terminal Iterative Learning Control

Simon V. Johansen[a,b], Martin R. Jensen[a], Bing Chu[c], Jan D. Bendtsen[b], Jesper Mogensen[a] and Eric Rogers[c]

[a]SKOV A/S, Hedelund 4, Glyngøre, Denmark
[b]Department of Control and Automation, Aalborg University, Denmark
[c]Department of Electronics and Computer Science, Southampton University, England

Abstract—*Broiler (chicken for meat production) growth maximization reduces the amount of feed, water and electricity required to produce a mature broiler where temperature control is one of the most influential factors. Iterative learning control provides a potential solution given the repeated nature of the production process, as it has been especially developed for systems that make repeated executions of the same finite duration task. Dynamic neural network models are used given the absence of mathematical models of the growth process. Traditional ILC is modified to maximize the terminal broiler weight and better cope with the uncertain nature of the data driven model. To evaluate the proposed algorithm in simulation, a heuristic broiler growth model based on the knowledge of a broiler application expert is formalized. This paper gives the first results on the application of optimization based iterative learning control.*

Keywords—*Iterative learning control, Biosystems, Neural networks*

Published in *Proceedings of Conference on Control Technology and Applications*, 2018.

Contents

1 Introduction

Broiler (chicken for meat production) growth maximization reduces the amount of feed, water and electricity required to produce a mature broiler. A major motivation factor in this area is that the global demand for poultry meat is predicted to increase by 19% between 2013-2015 and 2025 to 131.3 billion kg [OECD, 2016, pp. 127], of which broiler meat will represent the majority. Industrial state-of-the-art broiler production typically has 30-40,000 broilers per batch, produces 2050g broilers in 34 days from 42g newly hatched broilers, and employs ad libitum feeding and drinking strategies. Tight bounds on the production environment must be met to facilitate optimal growth, which requires manual tuning of each broiler shed by a broiler application expert. Active feed control is not practically feasible in state of the art broiler production as ad libitum feeding regimes are used; temperature control, on the other hand, is highly influential and practically feasible. Broiler production is mature in terms of data acquisition due to tight biosecurity and traceability requirements. This, in turn, drives the need to automatically optimize performance in a data driven framework by suitably designed temperature control. Given the absence of mathematical models of the growth process and the repeated nature of broiler production, Iterative Learning Control (ILC) provides a potential solution.

The first work on ILC is widely credited to [Arimoto et al., 1984] and is applicable to systems executing a finite duration operation over and over again. Once an execution is complete, the system resets to the starting location and the next execution can begin. In the literature, each execution is termed a trial (or a pass or iteration) and the duration of each trial is known as the trial length. Let $u_k[n]$ be the control input, $r[n]$ the reference, $y_k[n]$ the output and $e_k[n]$ the tracking error – all of appropriate dimensions for trial k and sample n. Then the goal of ILC is to force the tracking error $e_k[n]$ to converge to 0 in k such that $y_k[n] = r[n]$ for $k \to \infty$. Hence the control action is chosen to sequentially improve performance from trial-to-trial. A simple ILC law to compute the control input on trial $k+1$ is $u_{k+1}[n] = u_k[n] + \gamma e_k[n+\lambda]$ with the integer $0 < \lambda$ and the real scalar gain γ. This is known as phase-lead ILC due to the parameter $\lambda > 0$, i.e., information ahead of the current sample n is used. It is causal since trial k is completed but in a non-ILC setting this would otherwise be non-causal. The phase-lead ILC law can be designed without a model of the process dynamics by using empirical type rules to chose the parameters γ and λ. Another simple structure ILC law was the basis of the results reported in [Arimoto et al., 1984] (using so-called D-type ILC) and many designs using such control laws have been experimentally validated. A starting point for the literature is the survey papers [Bristow et al., 2006] and [Ahn et al., 2007].

Model based ILC will be required in many cases and this area has been the subject of much research for both linear and nonlinear dynamic models. These include algorithms based on minimizing a suitably chosen cost function with applications including gantry robots, e.g., [Paszke et al., 2013], additive

manufacturing, e.g., [Lim et al., 2017] and an extension to robotic-assisted stroke rehabilitation for the upper-limb with supporting clinical trials [Freeman et al., 2015]. This paper gives the first results on a new application of ILC to food production.

Dynamic neural networks (DNNs) are used because no mathematical models exist describing the dynamic relationship between climate conditions and broiler growth, and broiler production is a biological batch process which tends to be highly nonlinear and time varying. It is a nonlinear data driven modeling technique capable of approximating any non-discontinuous system – for a thorough introduction see [Du and Swamy, 2014] and [Haykin, 1994]. Such models have been successfully applied to model complex biological processes, of which non-control related applications include broiler growth forecasting [Johansen et al., 2017][Johansen et al., 2019], prediction of bioethanol production [Grahovac et al., 2016] and yeast fermentation modeling [Nasimi and Irani, 2014]. In this paper they are used as a basis for model based ILC design. Traditional ILC is modified to maximize the terminal broiler weight and better cope with the uncertain nature of the data driven model. To evaluate the proposed algorithm in simulation, a heuristic broiler growth model is formalized based on the experience and knowledge of a broiler application expert.

The heuristic broiler growth model is formulated in Section 2, the data driven broiler growth maximization algorithm is developed in Section 3 and tested in Section 4.

1.1 Notation

Let $u_k[n] \in \mathbb{R}^{N_u}$ denote a signal at trial k and sample n, then U_k is the super-vector version of $u_k[n]$ in the finite time interval between the first sample N_s and last sample N_e according to

$$U_k = \begin{bmatrix} u_k[N_s]^T & \cdots & u_k[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_u N_n} \tag{1}$$

with a total of $N_n = N_e - N_s + 1$ samples and $\tilde{U}_k = u_k[N_e]$ denotes the terminal super-vector signal. Let a denote a vector and A a square matrix, then $\|a\|_A = \sqrt{a^T A a}$ is the weighted euclidean norm and $\|a\| = \sqrt{a^T a}$ is the euclidean norm. Let b and c be sets, then $\#b$ denotes the cardinality of b and $b \backslash c = \{x \in b \mid x \notin c\}$ is the difference of b and c.

2 Heuristic Broiler Growth Model

The motivation behind developing the following heuristic broiler growth model is to test the data driven broiler growth optimization algorithm proposed in Section 3.3 in a simulation environment. Please note that only past growth model data, and not the growth model, is used for control synthesis, which would also be the case under real production conditions. Hence, it is only

necessary for the model to portray basic broiler growth behavior present in industrial *state-of-the-art* broiler production and is based on the experience and knowledge of a broiler application expert.

The model's primary objective is to test the algorithm's ability to iteratively learn a unique time series of broiler state dependent temperature inputs that maximize the terminal broiler growth, while simulating reduced growth for both negatively- and positively suboptimal temperature inputs. Such a broiler growth model can be represented with a discrete time dynamic nonlinear model by

$$x[n+1] = x[n] + T_s G(u[n], x[n]) \text{ with } x[N_s] = 0, \tag{2a}$$

$$z[n] = R_w(x[n]) \text{ and} \tag{2b}$$

$$y[n] = z[n] + q[n], \tag{2c}$$

where $x[n] \in \mathbb{R}_+$ is the broiler maturity in "effective growth days", $z[n] \in \mathbb{R}_+$ is the true broiler weight, $y[n] \in \mathbb{R}_+$ is the measured broiler weight, $u[n] \in \mathbb{R}$ is the temperature input and $T_s \in \mathbb{R}_+$ is the sampling interval in days. Note that under real production conditions the temperature input $u[n]$ is a reference for the climate control system, which for simplicity is assumed to achieve perfect tracking. In (2a), we introduce the smooth maturing rate function $G{:}\,\mathbb{R} \times \mathbb{R}_+ \to [\beta; 1]$ where $\beta \in [0; 1[$ is the worst case broiler growth rate due to temperature. Finally, $R_w{:}\,\mathbb{R}_+ \to \mathbb{R}_+$ is a smooth and strictly increasing function mapping the broiler maturity into broiler weight and $q[n]$ is the measurement noise.

The growth of the widely used ROSS 308 fast growing broiler strain is described by the manufacturer in [Aviagen, 2014, pp. 3] and equals

$$R_w(t) = \frac{-18.3\,t^3 + 2.2551\,t^2 + 2.9118\,t + 54.739}{1000} \tag{3}$$

where $R_w(t) \in \mathbb{R}_+$ is the broiler weight reference in kg and $t \in [0, 70]$ is the time in "effective growth days". Expressing broiler growth $R_w(x[n])$ in terms of the broiler maturity in "effective growth days" through $x[n]$ results in realistic growth behavior, as it accurately captures the nonlinear nature of broiler growth.

We find a modified normal distribution to be an appropriate expression for the maturing rate function G, as it has a unique maximum and the intuitive nature of the standard deviation can be exploited to design how sensitive G should be to temperature errors. We specifically use

$$G(u[n], x[n]) = \beta + (1 - \beta) \exp\left\{ \ln\left(\frac{\beta - \alpha}{\beta - 1}\right) \left[\frac{u[n] - \bar{u}(x[n])}{\sigma_u} \right]^2 \right\}, \tag{4}$$

where $\bar{u}(x[n])$ is the temperature maximizing G, $G(\bar{u}(x[n]), x[n]) = 1$, and $\sigma_u \in \mathbb{R}_+$ is the constant temperature sensitivity. The temperature sensitivity is the temperature input error, $u[n] - \bar{u}(x[n])$, resulting in decreased maturing rate of α – equivalent of $G(\bar{u}(x[n]) \pm \sigma_u, x[n]) = 1 - \alpha$ with $\alpha \in]0; 1-\beta[$. The

intention is to allow the broiler application expert to heuristically specify the decreased growth rate for a specific temperature error. On Figure 1 the components of the maturing rate function G is visualized. For a more accurate temperature sensitivity, the broilers' feathering and ability to regulate their own body temperature should also be considered – possibly making σ_u time and state dependent. However, this is left as a subject for future work.

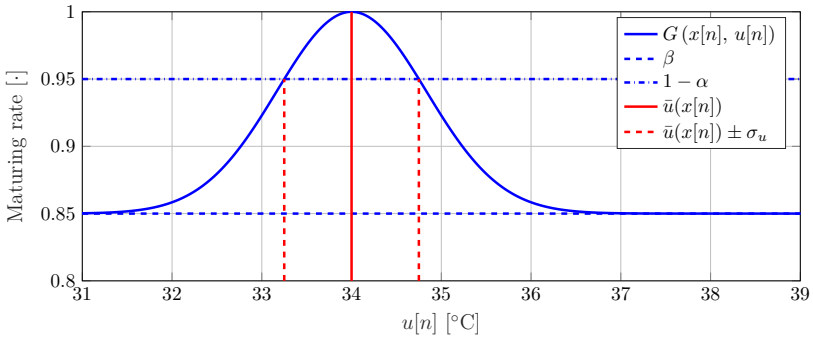


Figure 1: Visualization of the maturing rate function $G(x[n], u[n])$ for $x[n] = 0$ with worst case broiler growth rate $\beta = 0.85$, $\alpha = 0.05$, maximizing input $\bar{u}(x[n]) = 34\,[°\mathrm{C}]$ and temperature error sensitivity $\sigma_u = 0.75\,[°\mathrm{C}]$.

The optimal temperature profile is unknown in the industry, but typical temperature profiles for the ROSS 308 fast growing broiler transition almost linearly between the initial temperature of $\bar{u}_s = 34$ °C at day $t_s = 0$ to $\bar{u}_e = 21$ °C at day $t_e = 34$. This corresponds to a temperature drop of $(\bar{u}_e - \bar{u}_s)$, which we model as being proportional with the maturity $x[n]$ according to

$$\bar{u}(x[n]) = \bar{u}_s + \Delta T x[n] \text{ with } \Delta T = \frac{\bar{u}_e - \bar{u}_s}{t_e - t_s}. \tag{5}$$

Consequently, the optimal temperature at sample n depends on $x[n-1]$, which depends on all prior inputs.

Inspecting G shows that $x[n]$ is maximized by the unique input $\bar{u}(x[n])$ satisfying

$$G(u[n], x[n]) < G(\bar{u}(x[n]), x[n]) = 1 \quad \forall u[n] \neq \bar{u}(x[n]).$$

With $\beta \leq G \leq 1$ the largest possible maturity $\bar{x}[n]$ equals

$$\bar{x}[n] = \max\{x[n]\} = T_s \sum_{i=1}^{n} \max\{G(u[i], x[i])\} = n\,T_s.$$

As R_w is strictly increasing, the largest possible true broiler weight $\bar{z}[n]$ equals

$$\bar{z}[n] = \max\{z[n]\} = \max\{R_w(x[n])\} = R_w(\max\{x[n]\}) = R_w(n\,T_s).$$

This ensures that suboptimal control results in suboptimal growth, as expected in real broiler production where either a too low or high temperature results in decreased broiler growth.

On Figure 2 the behavior of the broiler model is depicted for different temperature inputs. Interestingly, if the temperature is positively suboptimal, e.g., $u[n] = \bar{u}(\bar{x}[n]) + 1\,^{\circ}\mathrm{C}$, then G converges to 1 for $n \to \infty$, and if the temperature is negatively suboptimal, e.g., $u[n] = \bar{u}(\bar{x}[n]) - 1\,^{\circ}\mathrm{C}$, then G converges to β for $n \to \infty$. This is caused by the decreasing $\bar{u}(x[n])$ for increasing $x[n]$ – supported by the fact that broiler farmers tend to use positively suboptimal temperatures.

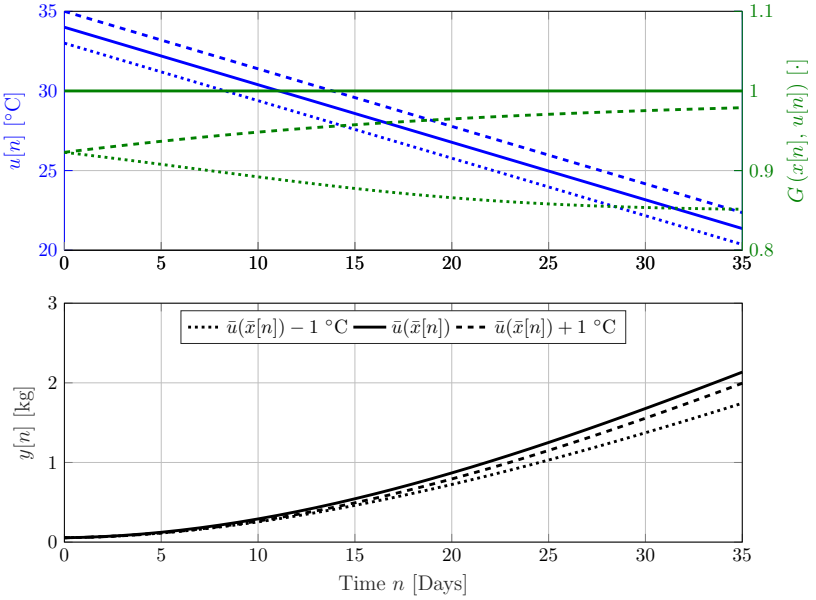


Figure 2: Visualization of the broiler growth model with different inputs. The top plot depicts the maturing rate function $G(u[n], u[n])$ as a function of the input $u[n]$ and the bottom plot depicts the output $y[n]$. The model settings equal that of Figure 1 with $T_s = 1$ day.

Lastly, when evaluating the ILC algorithm proposed in Section 3.3, an analytical linear terminal super-vector broiler growth model of $\tilde{Z}_k \in \mathbb{R}^{N_y}$ is required. It is obtained by linearizing (2) along the trajectory of $U_k \in \mathbb{R}^{N_u N_n}$ using the first order Taylor expansion according to

$$\tilde{Z}_k(U) \approx \tilde{Z}_k(U_k) + \tilde{P}_k\,(U - U_k) = \tilde{P}_k U + \tilde{K}_k \tag{6}$$

with

$$\tilde{P}_k = \left.\frac{d\,\tilde{Z}_k(U_k)}{d\,U_k^T}\right|_{U_k} \text{ and } \tilde{K}_k = \tilde{Z}_k(U_k) - \tilde{P}_k U_k,$$

where $\tilde{P}_k \in \mathbb{R}^{N_y \times N_u N_n}$ is the terminal model matrix, and $\tilde{K} \in \mathbb{R}^{N_y}$ is the terminal output constant vector unrelated to the input $U \in \mathbb{R}^{N_u N_n}$.

3 Broiler Growth Optimization

3.1 Terminal Iterative Learning Control (TILC)

TILC is a method that can be applied to a repeating process, with the aim of iteratively learning the input sequence $U_k \in \mathbb{R}^{N_u N_n}$ such that the terminal process output $\tilde{Y}_k(U_k) \in \mathbb{R}^{N_y}$ tracks the desired terminal reference $\tilde{R} \in \mathbb{R}^{N_y}$ denoted by

$$\lim_{k \to \infty} \tilde{Y}_k(U_k) = \tilde{R}. \tag{7}$$

This problem can be solved using constrained Norm Optimal Point-To-Point ILC, which aims at tracking the output at specific samples. As TILC only aims at tracking the terminal output, TILC is a specialization of Point-To-Point ILC. Adapting the constrained Norm Optimal Point-To-Point ILC algorithm 1 in [Chu et al., 2015] to the special case of TILC yields the problem

$$U_{k+1} = \underset{U \in \Omega}{\arg\min} \, \|\tilde{E}_k(U)\|^2_{W_{\tilde{E}}} + \|U - U_k\|^2_{W_{\Delta U}} \tag{8a}$$

subject to

$$\tilde{E}_k(U) = \tilde{R} - \tilde{Y}_k(U) \text{ and} \tag{8b}$$

$$\tilde{Y}_k(U) = \tilde{P}U + \tilde{K}, \tag{8c}$$

where Ω is the set of valid inputs, $0 < W_{\tilde{E}} \in \mathbb{R}^{N_y \times N_y}$ is the positive definite tracking error cost matrix, $0 < W_{\Delta U} \in \mathbb{R}^{N_u N_n \times N_u N_n}$ is the positive definite input change cost matrix, and $\tilde{E}_k(U) \in \mathbb{R}^{N_y}$ is the terminal tracking error given by (8b). The super-vector model used for control synthesis by the ILC is given by (8c), where $\tilde{P} \in \mathbb{R}^{N_y \times N_u N_n}$ is the terminal system matrix, and $\tilde{K} \in \mathbb{R}^{N_y}$ represents terminal effects unrelated to the input $U \in \mathbb{R}^{N_u N_n}$. The intuition behind (8) is to reduce the terminal tracking error by finding an input in the neighborhood of U_k that norm-optimally minimizes the cost function given by (8a).

If perfect tracking is not feasible, $\tilde{Y}_k(U) \neq \tilde{R} \quad \forall U \in \Omega$; then [Chu et al., 2015] shows that the input monotonically converges to

$$\lim_{k \to \infty} U_{k+1} = \underset{U \in \Omega}{\arg\min} \, \|\tilde{R} - \tilde{P}U - \tilde{K}\|^2_{W_{\tilde{E}}}, \tag{9}$$

equivalent of the algorithm converging to the smallest possible tracking error.

3.2 Data Driven Model

The objective of the data driven model is to enable control synthesis without a mathematical broiler growth model, by synthesizing $\tilde{P}$ and $\tilde{K}$ from (8c) using past production data. Using a nonlinear discrete time data driven model we intend to capture the broiler growth dynamic using data from the past N_b trials, $\{\{U_{k-N_b+1}, Y_{k-N_b+1}\}, \cdots, \{U_k, Y_k\}\}$, of which the N_b data indexes are conveniently denoted by

$$D_k = \{k - N_b + 1, \cdots, k\}. \tag{10}$$

For data driven model synthesis at trial k, data from the trial indexes denoted by D_{k-1} are required. Trial data prior to the first trial, k<1, are denoted *preliminary* trials, e.g., $\{U_{-2}, Y_{-2}\}$. Hence, a total of N_b preliminary trials are required for model synthesis for the first trial, $k = 1$, denoted by the indexes $D_0 = \{1 - N_b, \cdots, 0\}$. The data driven model is choosen to be a NARMAX type neural network model with N_l input and output lags, a single hidden layer with N_N neurons and hyperbolic tangent activation function in the hidden layer according to

$$\hat{y}_k[n+1 \mid \mathcal{W}, s] = W^{\mathrm{o}} \tanh\big(\mathcal{X} + \theta^{\mathrm{h}}\big) + \theta^{\mathrm{o}} \tag{11}$$

with

$$\mathcal{X} = \sum_{i=0}^{N_l-1} W^{\mathrm{h}}_{y,i} \hat{y}_k[n-i \mid \mathcal{W}, s] + W^{\mathrm{h}}_{u,i} u_k[n-i],$$

where $W^{\mathrm{o}} \in \mathbb{R}^{N_y \times N_N}$, $\mathcal{X} \in \mathbb{R}^{N_N}$, $\theta^{\mathrm{o}} \in \mathbb{R}^{N_N}$, $W^{\mathrm{h}}_{y,i} \in \mathbb{R}^{N_N \times N_y}$, $W^{\mathrm{h}}_{u,i} \in \mathbb{R}^{N_N \times N_u}$ and $\theta^{\mathrm{h}} \in \mathbb{R}^{N_N}$ are model parameters abstractly stored in $\mathcal{W}$, while $\hat{y}_k[n \mid \mathcal{W}, s]$ is the model output at sample n, initialized at sample s with model weights $\mathcal{W}$. Initialization in this case refers to

$$\hat{y}_k[n \mid \mathcal{W}, s] = y_k[n] \quad \forall n \le s, \tag{12}$$

where n is implicitly lower bounded by the start sample N_s, $N_s \le n$, for both $y_k[n]$ and $u_k[n]$. The model weights are found through training by solving

$$\mathcal{W}(B) = \arg\min_{\mathcal{W}} \sum_{b \in B \setminus \min\{B\}} \frac{J_b(\mathcal{W})}{\#B - 1} \tag{13}$$

with

$$J_b(\mathcal{W}) = \bar{\alpha}\|\mathcal{W}\|^2 + \sum_{i=1}^{N_S} \sum_{n=S_i}^{N_e} \frac{\|y_b[n] - \hat{y}_b[n \mid \mathcal{W}, S_i]\|^2}{N_y(N_e - S_i + 1)}$$

where B is a set of batch-indexes used for training and $S = \{S_1, \cdots, S_{N_S}\}$ is the set of N_S initialization locations, which was found to speed up training

as described in [Johansen et al., 2019]. It is minimized using the Levenberg-Marquardt algorithm with early stopping applied on the oldest batch index in B, denoted by $\min\{B\}$, according to $J_{\min\{B\}}(\mathcal{W})$ to prevent overtraining. The regularization constant $\bar{\alpha} \in \mathbb{R}_+$ is found iteratively through Bayesian regularization to prevent overfitting. The model weights $\mathcal{W}$ are initialized using the Nguyen Widrow initialization scheme. For detailed information regarding the training see [Johansen et al., 2017] and [Johansen et al., 2019].

As (13) is not a convex optimization problem, the weights $\mathcal{W}(B)$ are not guaranteed to be the global minimum. To decrease the probability of ending up in a local minimum, the ensemble mean of N_m models trained with different initial model weights is used. The ensemble data driven model simulated from sample N_s with data from batch b, $\{Y_b, U_b\}$, equals

$$\hat{y}_{k,b}[n] = \frac{1}{N_m} \sum_{l=1}^{N_m} \hat{y}_b[n \mid \mathcal{W}_l(D_k \backslash b), N_s], \tag{14}$$

where $\mathcal{W}_l(D_k \backslash b)$ is the l'th training of $\mathcal{W}(D_k \backslash b)$ with the batch indexes $D_k \backslash b$ to separate training data and simulation data. The terminal super-vector ensemble data driven model required for (8) is obtained by linearizing (14) along the trajectory of U_b using the first order Taylor expansion

$$\tilde{Y}_k(U) \approx \hat{\tilde{Y}}_{k,b} + \hat{\tilde{P}}_{k,b}(U - U_b) = \hat{\tilde{P}}_{k,b} U + \hat{\tilde{K}}_{k,b} \tag{15}$$

with

$$\hat{\tilde{P}}_{k,b} = \left. \frac{d\hat{\tilde{Y}}_{k,b}}{dU_b^T} \right|_{U_b} \text{ and } \hat{\tilde{K}}_{k,b} = \hat{\tilde{Y}}_{k,b}(U_b) - \hat{\tilde{P}}_{k,b} U_b$$

where $U \in \mathbb{R}^{N_u N_n}$ is the super-vector input used in (8c). The data driven model is retrained for every k and b.

3.3 Data Driven TILC Broiler Optimization

The objective of the method is to obtain the maximal terminal broiler weight, which is unknown in broiler production. One reason for this is that artificial genetic selection progressively increases the growth rate. To cope with this the reference is redefined to

$$\tilde{R}_k = \tilde{Y}_k(U_k) + \mathcal{R}, \tag{16}$$

where $\mathcal{R} \in \mathbb{R}^{N_y}$ is a strictly positive trial-independent maximization constant vector, and this method is denoted *maximizing reference*. As $\tilde{E}_k(U_k) = \tilde{R}_k - \tilde{Y}_k(U_k) = \mathcal{R}$ is constant, zero tracking error is unobtainable by construction. Assuming that $\tilde{Y}_k(U_k)$ is upper bounded by $\tilde{Y}_{\max} \in \mathbb{R}^{N_y}$ in combination with (9), then we hope that

$$\lim_{k \to \infty} \tilde{Y}_k(U_k) = \tilde{Y}_{\max} \text{ and } \lim_{k \to \infty} \tilde{R}_k = \tilde{Y}_{\max} + \mathcal{R}. \tag{17}$$

However, as broiler growth is a nonlinear process, a local maximum might be obtained instead of $\tilde{Y}_{\max}$.

In the following the notion of the best recent trial index κ_k is required and for $\tilde{Y}_i(U_i) \in \mathbb{R}_+^{N_y}$ it is defined by

$$\kappa_k = \underset{i \in [\max(k-N_b,0);k]}{\arg\max} \|\tilde{Y}_i(U_i)\|_{W_{\tilde{E}}}, \tag{18}$$

and serves as a feasible substitute for the *true best recent trial index* given by

$$\underset{i \in [\max(k-N_b,0);k]}{\arg\min} \|\tilde{Y}_{\max} - \tilde{Y}_i(U_i)\|_{W_{\tilde{E}}}.$$

Note that i is lower bounded by 0, which equals the most recent preliminary trial and that the particular expression of (18) is application-dependent. To cope with the uncertain nature of the data driven model given by (15) the TILC algorithm is modified into a descent type algorithm, denoted *anchoring*, by solving

$$U_{k+1} = \underset{U \in \Omega_{k+1}}{\arg\min} \|\tilde{E}_{\kappa_k}(U)\|_{W_{\tilde{E}}}^2 + \|U - U_{\kappa_k}\|_{W_{\Delta U}}^2 \tag{19a}$$

subject to (16), (18),

$$\tilde{E}_{\kappa_k}(U) = \tilde{R}_{\kappa_k} - \tilde{Y}_{\kappa_k}(U) \text{ and} \tag{19b}$$

$$\tilde{Y}_{\kappa_k}(U) = \hat{\tilde{P}}_{k,\kappa_k} U + \hat{\tilde{K}}_{k,\kappa_k} \tag{19c}$$

where $\Omega_{k+1} \in \mathbb{R}^{N_u N_n}$ is the set of valid trial dependent inputs. Note that if the input U_{k+1} does not decrease the error in (18), then the data driven model is not sufficiently accurate in the neighborhood of U_{κ_k}, i.e. $\hat{\tilde{P}}_{k,\kappa_k} \neq \tilde{P}_k$. Therefore U_{k+1} is rejected and U_{k+2} is calculated in the neighborhood of $U_{\kappa_{k+1}} = U_{\kappa_k}$ instead of U_{k+1}. This effectively ensures that the algorithm keeps exploring the neighborhood of the recent best trial input U_{κ_k} until the data driven model is sufficiently accurate to maximize the terminal output norm in (18), as the data driven model always uses the most recent data from the last N_b trials. Consequently, if the data driven model $\hat{\tilde{P}}_{k,\kappa_k}$ is identical to the analytical model $\tilde{P}_{\kappa_k}$ under ideal conditions and constant reference, then $\kappa_k = k$ as $\tilde{E}_k$ is monotonically decreasing in k.

The computable solution of (19) equals

$$U_{k+1} = U_{\kappa_k} + \underset{\Delta U \in \Omega_{k+1} - U_{\kappa_k}}{\arg\min} \frac{1}{2}\|\Delta U\|_{Q_1}^2 + Q_2^T \Delta U \tag{20}$$

with

$$Q_1 = 2\big(\hat{\tilde{P}}_{k,\kappa_k}^T W_{\tilde{E}} \hat{\tilde{P}}_{k,\kappa_k} + W_{\Delta U}\big) \text{ and}$$
$$Q_2 = -2\hat{\tilde{P}}_{k,\kappa_k}^T W_{\tilde{E}} \tilde{E}_{\kappa_k}(U_{\kappa_k})$$

where $\Delta U = U - U_{\kappa_k}$ and results in an algorithm of the form $U_{k+1} = F(U_{\kappa_k}, \tilde{E}_{\kappa_k}(U_{\kappa_k})) = F(U_{\kappa_k}, \tilde{R}_{\kappa_k} - \tilde{Y}_{\kappa_k}(U_{\kappa_k}))$ that facilitates feedback though the measured terminal output through $\tilde{Y}_{\kappa_k}(U_{\kappa_k})$ and $\hat{\tilde{P}}_{k,\kappa_k}$. If combined with maximizing reference then $\tilde{E}_k(U_{\kappa_k}) = \mathcal{R}$ and $\tilde{Y}_{\kappa_k}(U_{\kappa_k})$ is only used indirectly through $\hat{\tilde{P}}_{k,\kappa_k}$. This problem can be solved using standard quadratic programming solvers, such as Matlab's `quadprog`.

4 Simulation Study

4.1 Simulation Study Description

The objective of this simulation study is to investigate the ability of different configurations of the data driven optimization algorithm given by (19) to maximize the terminal broiler weight $\tilde{Y}_k$ of the heuristic broiler growth model given by (2). We specifically investigate the performance impact of:

1. using the data driven model $\hat{\tilde{P}}_{k,\kappa_k}$ for control synthesis from (15), indicated by (D), compared to unrealistically using the analytical super-vector model $\tilde{P}_{\kappa_k}$ for control synthesis from (6), indicated by (I), as depicted on Figure 3b.

2. using anchoring though κ_k from (18), indicated by (A), compared to disabling it by forcing $\kappa_k = k$, indicated by $(\cdot)$, as depicted on Figure 3c.

3. using maximizing reference from (16), indicated by (MR), compared to unrealistically using the true analytical maximum given by

$$\tilde{R}_k = \tilde{Y}_{\max} = \bar{z}[N_e], \tag{21}$$

   indicated by $(\cdot)$, as depicted on Figure 3d.

This results in a total of 8 different test configurations, of which some are depicted on Figure 3. Each test is repeated 10 times and the mean true absolute terminal error, $|\tilde{Z}_k - \tilde{R}_{\max}|$, is used for evaluation.

4.2 Method and Model Configuration

The heuristic broiler growth model in Section 2 is simulated between the initial sample $N_s = 0$ and the terminal sample $N_e = 35$ with a sample interval of $T_s = 1$ day, and is heuristically configured as follows. We use $\beta = 0.85$ as the worst case maturing rate, as feed and water consumption are the dominating factors, while correct temperature control is regarded as a catalyst. We use $\alpha = 0.05$ and $\sigma_u = 0.75\,[^\circ\mathrm{C}]$ as we regard it as a good overall sensitivity throughout the lifespan of a broiler.

The measurement noise $q[n]$ is found by analyzing the frequency spectrum of production data. As broiler growth is an under-damped process the "true"

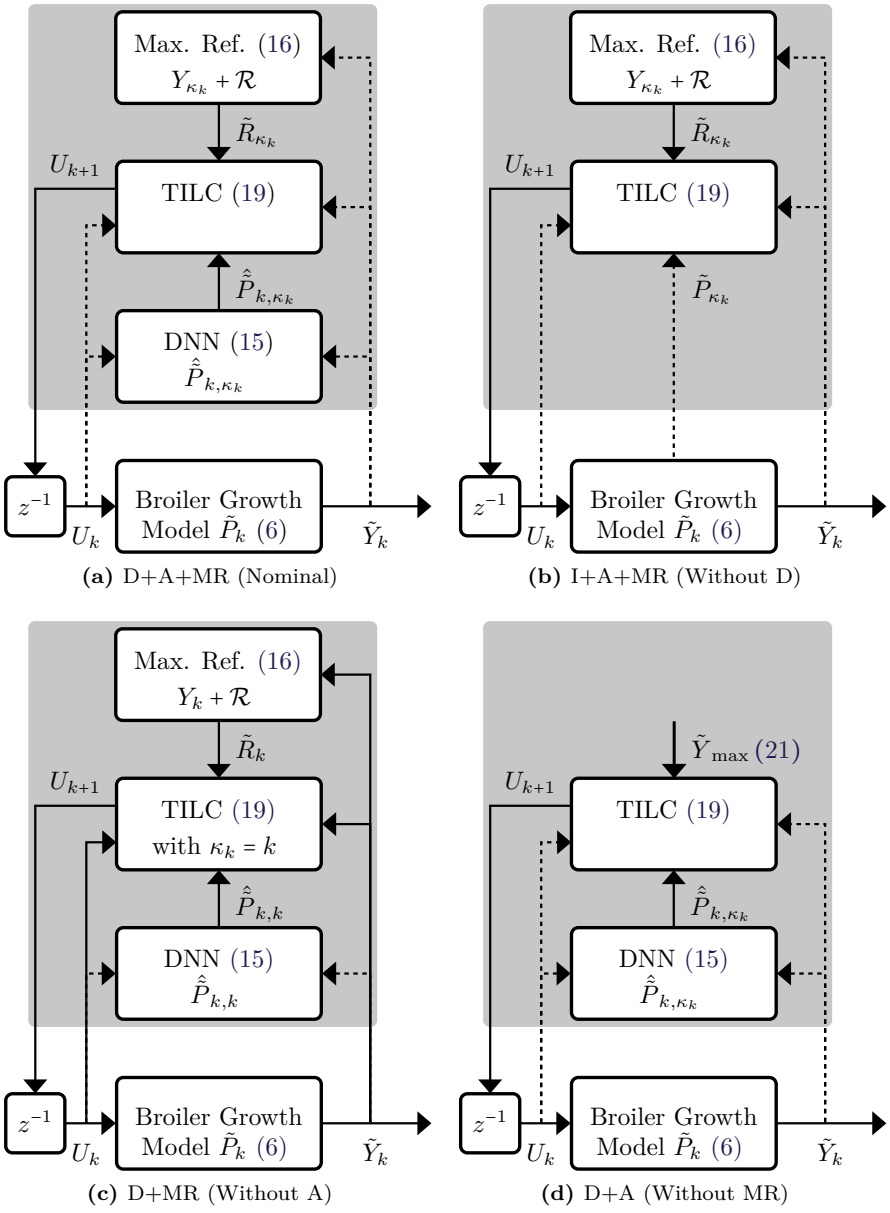


Figure 3: Illustration of some of the configurations of the broiler growth optimization algorithm tested in Section 4. The shaded area denotes the controller, z^{-1} denotes a unit delay, a dashed signal contains information from the last N_b trials, $\{k - N_b + 1, \cdots, k\}$, and a non-dashed signal only contains information from trial k. See Section 4.1 for detailed explanation.

broiler weight is approximated by a second order polynomial $\hat{y}_{k,\text{pol},2}$ between day 3 and 21, where the weight measurement y_k is expected to be the most reliable. The fit errors, $y_k - \hat{y}_{k,\text{pol},2}$, of 31 batches are depicted on the top plot of Figure 4 and is assumed to be measurement noise. Subtracting the mean, concatenating all the fit errors and applying the FFT produces the bottom magnitude plot. As this is not a standard distribution, random realizations of $q[n]$ with identical magnitude are obtained by randomly rotating the phases of the FFT and applying the inverse discrete Fourier transform. Some realizations of $q[n]$ are depicted on the top plot of Figure 4. For more information on this technique see [Prichard and Theiler, 1994].

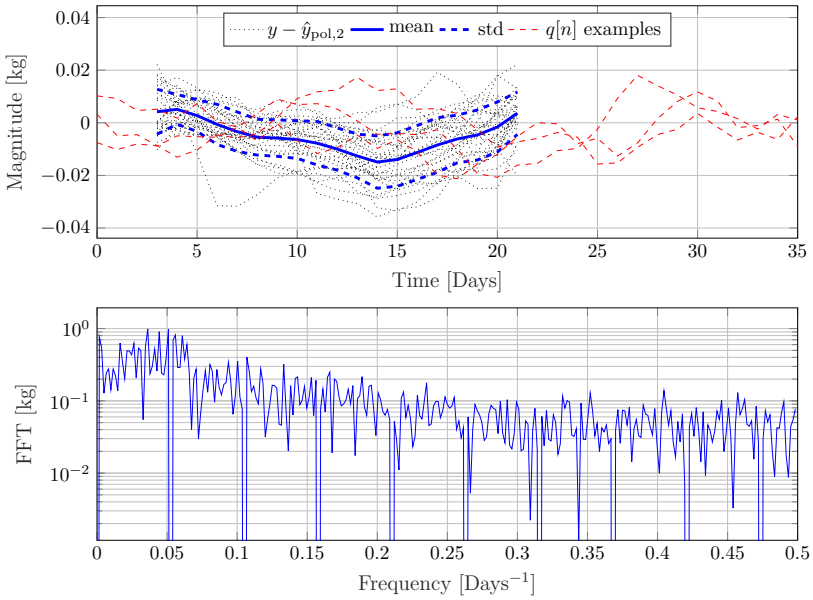


Figure 4: Visualization of the noise $q[n]$.

The data driven modeling method in Section 3.2 is heuristically generated with $N_m = 20$ ensemble models using $N_b = 10$ training batches, $N_l = 3$ input and output lags, $N_N = 7$ neurons in the hidden layer and with $N_S = 5$ initialization locations at samples $S = \{0, 7, 14, 21, 28\}$. The preliminary N_b trials required for training are generated using the positive input $u[n]$ resulting in 5% decreased maturing rate, $G(u[n], x[n]) = 0.95$, of which an example is depicted on Figure 5. To ensure identical initial input U_0 for all the tests, the most recent preliminary trial $k = 0$ does not have any added noise. Hence, the objective is to increase the terminal broiler weight $\tilde{Y}_k$ by 0.162 kg. White noise with standard deviation of 0.3 °C is added to the remaining $N_b - 1$ preliminary trials, $\{1 - N_b, \cdots, -1\}$. This is considered realistic, as most broiler farmers tend to use a too high temperature with little variations trial-to-trial.

Fast convergence conditions for the data driven TILC broiler optimization

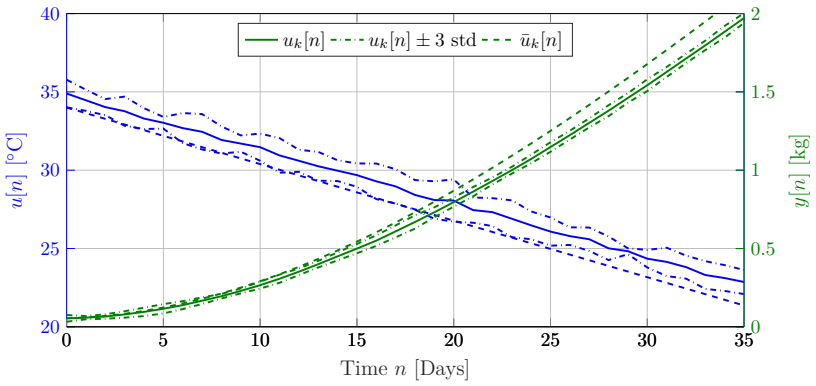


Figure 5: Visualization of 10 preliminary trial data.

algorithm are obtained by using a maximization constant of $\mathcal{R} = 0.035$ kg, terminal tracking error cost and input change cost of $W_{\tilde{E}} = (0.035\,\mathrm{kg})^{-2}$ and $W_{\Delta U} = \mathrm{diag}\big([1\,^\circ\mathrm{C}]^{-2}, \cdots, [1\,^\circ\mathrm{C}]^{-2}\big)$. The permitted temperature change is restricted to avoid large input fluctuations caused by data driven modeling errors in $\hat{\tilde{P}}_{k,\kappa_k}$. The valid input space Ω_{k+1} is therefore defined by

$$\omega_{k+1}[n] = \{u \,|\, u_{\kappa_k}[n] - \gamma \leq u \leq u_{\kappa_k}[n] + \gamma\} \tag{22}$$

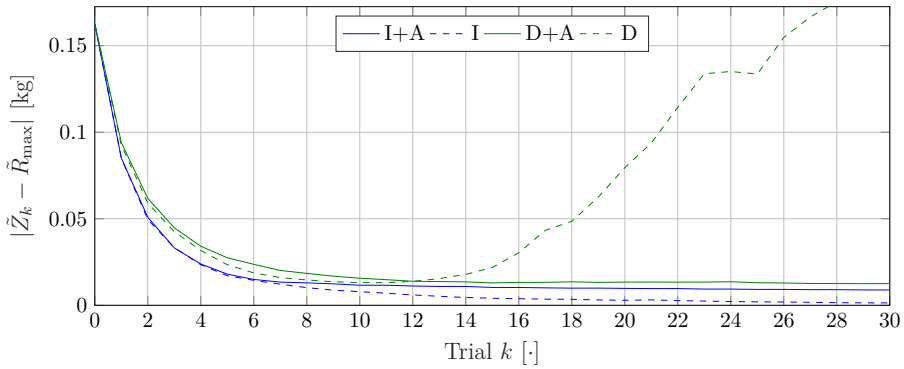
where $u \in \mathbb{R}$ is the input and $\gamma = 0.5$ °C is the lower and upper temperature change bound. Note that this does not restrict the permitted input space Ω_{k+1} for $k \to \infty$ as it change with $u_{\kappa_k}[n]$.

4.3 Simulation Results

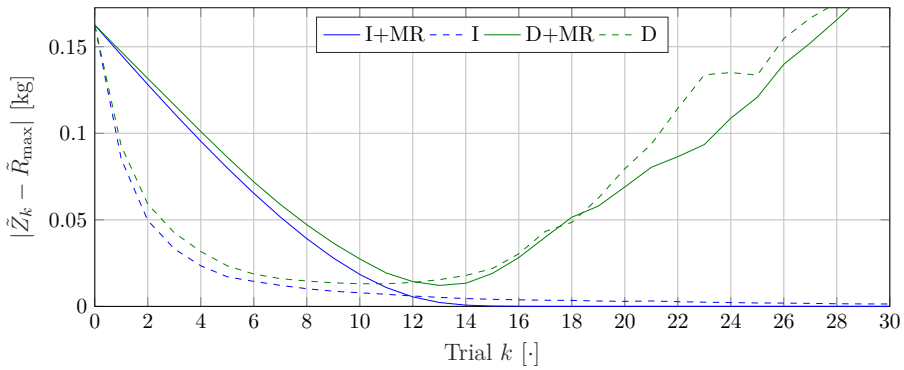
From Figure 6a we conclude that anchoring is beneficial in conjunction with the data driven model, as D diverges while D+A converges monotonically. This makes anchoring superior under data driven modeling conditions. However, anchoring appears to result in significantly slower convergence for small errors under ideal model conditions, as I approaches 0 error faster than I+A after having a comparable convergence for the first 6 trials.

From I+MR on Figure 6b we conclude that using the maximizing reference produces a constant improvement until $|\tilde{Z}_k - \tilde{R}_{\max}| < \mathcal{R}$, where the convergence slows down but approaches 0 error in finite time. If the maximizing reference is not used as in I, then the error is monotonically converging to 0 with decreasing step sizes. MR does not improve convergence conditions with a data driven model, as D+MR and D does not converge to 0 error.

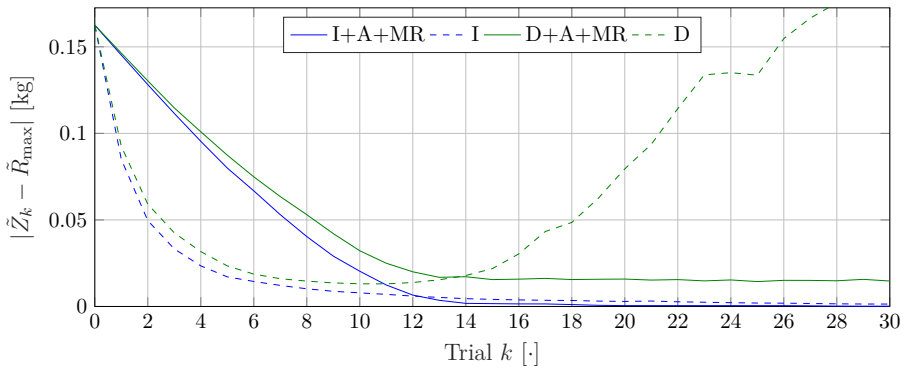
Using both MR and A, as depicted on Figure 6c, we conclude that D+A+MR is the best performing implementable configuration of the algorithm, as D does not converge despite I and I+A+MR having superior performance. Using MR in combination with D might even be preferred, as the output and input changes



(a) Anchoring



(b) Maximizing Reference



(c) Anchoring and Maximizing Reference

Figure 6: Simulation results – see Section 4 for detailed explanations.

evenly which decrease the risk of moving too far outside the valid data driven model region.

5 Conclusions and Future Work

In this paper a heuristic broiler growth model has been formulated and used to investigate the performance of a data driven optimization based ILC law in simulation. Traditional ILC is modified to maximize the terminal broiler weight and better cope with the uncertain nature of the data driven model. The heuristic broiler growth model is based on experience from a broiler application expert and approximates the dynamic behavior between broiler growth and temperature. Extensive simulation based studies confirm the potential of this approach and future work should include enhancing the broiler growth model to account for feed consumption and extending the control design to also minimize the feed consumption, but most importantly it should include experimental verification.

References

H.-S. Ahn, Y. Chen, and K. Moore. "Iterative learning control: Brief survey and categorization". *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1099 –1121, 2007.

S. Arimoto, S. Kawamura, and F. Miyazaki. "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems". In *The 23rd IEEE Conference on Decision and Control*. IEEE, dec 1984. doi:10.1109/cdc.1984.272176.

Aviagen. *Ross 308 Broiler: Performance Objectives*, 2014. URL: `http://en.aviagen.com/assets/Tech_Center/Ross_Broiler/Ross-308-Broiler-PO-2014-EN.pdf`.

D. A. Bristow, M. Tharayil, and A. G. Alleyne. "A survey of iterative learning control". *Control Systems, IEEE*, 26(3):96 – 114, 2006.

B. Chu, C. T. Freeman, and D. H. Owens. "A novel design framework for point-to-point ILC using successive projection". *IEEE Transactions on Control Systems Technology*, 23(3):1156–1163, may 2015. doi:10.1109/tcst.2014.2356931.

K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer Science + Business Media, 2014. doi:10.1007/978-1-4471-5571-3.

C. T. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes, and K. L. Meadmore. *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation*. Springer, 2015. ISBN 978-1-4471-6725-9. doi:10.1007/978-1-4471-6726-6.

J. Grahovac, A. Jokić, J. Dodić, D. Vučurović, and S. Dodić. "Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks". *Renewable Energy*, 85:953–958, jan 2016. doi:10.1016/j.renene.2015.07.054.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Broiler weight forecasting using dynamic neural network models with input variable selection". *Journal of Computers and Electronics in Agriculture*, 2019. doi:10.1016/j.compag.2018.12.014.

I. Lim, D. J. Hoelzle, and K. L. Barton. "A multi-objective iterative learning control approach for additive manufacturing applications". *Control Engineering Practice*, 64(11):74–87, 2017.

R. Nasimi and R. Irani. "Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks". *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14):1604–1611, may 2014. doi:10.1080/15567036.2011.592903.

OECD. *OECD-FAO Agricultural Outlook 2016-2025*. OECD Publishing, jul 2016. doi:10.1787/agr_outlook-2016-en.

W. Paszke, E. Rogers, K. Galkowski, and Z. Cai. "Robust finite frequency range iterative learning control design and experimental verification". *Control Engineering Practice*, 21(10):1310–1320, 2013.

D. Prichard and J. Theiler. "Generating surrogate data for time series with several simultaneously measured variables". *Physical Review Letters*, 73(7):951–954, aug 1994. doi:10.1103/physrevlett.73.951.

Paper F

Broiler FCR optimization using norm optimal terminal iterative learning control

Simon V. Johansen[a,b], Martin R. Jensen[a], Bing Chu[c], Jan D. Bendtsen[b], Jesper Mogensen[a] and Eric Rogers[c]

[a]SKOV A/S, Hedelund 4, Glyngøre, Denmark
[b]Department of Control and Automation, Aalborg University, Denmark
[c]Department of Electronics and Computer Science, Southampton University, England

Abstract—*Broiler feed conversion rate optimization reduces the amount of feed, water, and electricity required to produce a mature broiler, where temperature control is one of the most influential factors. Iterative learning control provides a potential solution given the repeated nature of the production process, as it has been especially developed for systems that make repeated executions of the same finite duration task. Dynamic neural network models provide a basis for control synthesis, as no first-principle mathematical models of the broiler growth process exist. The final feed conversion rate at slaughter is one of the primary performance parameters for broiler production, and it is minimized using a modified terminal iterative learning control algorithm in this work. This algorithm is evaluated in simulation, using a heuristic broiler growth model based on the knowledge of a broiler application expert, and experimentally on a* state-of-the-art *broiler house that produces around 40,000 broilers per batch.*

Keywords—*Iterative learning control, Biosystems, Neural networks*

Under review in *IEEE Transactions on Control Systems Technology.*

Contents

1 Introduction

The global demand for poultry meat is predicted to increase by 18% between 2015-2017 and 2027 to 139 billion kg [OECD, 2018, pp. 37], of which broiler (i.e., a chicken that is bred and raised specifically for meat production) meat will represent the majority. Industrial state-of-the-art broiler production typically has 30-40,000 broilers per batch, produces 2050g broilers in 34 days from 42g newly hatched broilers and employs ad libitum feeding and drinking strategies, i.e., unrestricted access to feed and water. Broiler feed conversion rate (FCR) optimization reduces the amount of feed, water and electricity required to produce a mature broiler.

Tight bounds on the production environment must be met to enable optimal growth, which requires manual tuning of each broiler house by a broiler application expert. Active feed control is not practically feasible in state of the art broiler production as ad libitum feeding regimes are used. Temperature control is, however, highly influential and practically feasible. Broiler production is mature in terms of data acquisition due to tight biosecurity and traceability requirements. This, in turn, drives the need to automatically optimize performance in a data driven framework by suitably designed temperature control. In this paper, a design based on combining Iterative Learning Control (ILC) and Dynamic neural network (DNN) modeling is developed and evaluated in both simulation and implementation.

The development of ILC was motivated by the many processes that repeat the same finite duration task over and over again, e.g., a gantry robot undertaking a "pick and place" task. Each execution is commonly termed a trial or pass and the finite duration is known as the pass or trial length. Once a trial is completed, the system resets to the starting location and the next trial can begin, exactly as in broiler production. Moreover, all data recorded during the previous trial is available for use in computing the control input for the next trial with the overall aim of improving performance trial-to-trial.

See the survey papers [Bristow et al., 2006] and [Ahn et al., 2007] for a good starting point for ILC literature, where [Arimoto et al., 1984] is one of the first published papers on the subject. ILC range from simple structure laws, such as phase-lead, that can be tuned without the use of a model, to advanced model based designs for linear and nonlinear dynamics. Mature ILC application areas with experimental validation and implementation include [Paszke et al., 2013], additive manufacturing, e.g., [Lim et al., 2017] and an extension to robotic-assisted stroke rehabilitation for the upper-limb with supporting clinical trials [Freeman et al., 2015].

Model based ILC is required for broiler FCR optimization since the broiler growth process itself is highly nonlinear and time varying. This paper uses nonlinear data driven modeling in the form of dynamic neural networks to model the dynamic relationship between climate conditions and broiler growth. See, e.g., [Du and Swamy, 2014] and [Haykin, 1994] for background information on neural networks. Such models have been successfully applied to model complex

biological processes, of which non-control related applications include broiler growth forecasting [Johansen et al., 2017][Johansen et al., 2019b], prediction of bioethanol production [Grahovac et al., 2016] and yeast fermentation modeling [Nasimi and Irani, 2014].

This paper gives the first results on a new application of ILC to food production. In particular, ILC is modified to minimize the terminal broiler FCR in the presence of the uncertain nature of the data driven DNN model. To evaluate the new design in simulation, a heuristic broiler growth model is formalized based on the experience and knowledge of a broiler application expert, which is then analyzed to provide FCR optimization guidelines. Preliminary results of parts of this work were given in [Johansen et al., 2018]. The results in this paper differ substantially by including cumulative feed consumption output in the heuristic model, measurement weight bias compensation as investigated in [Johansen et al., 2019a], and experimental results are finally provided from a *state-of-the-art* broiler production facility.

The paper is organized as follows. The development of a heuristic broiler model and the broiler FCR minimization problem is described in Section 2. Terminal ILC is then introduced and applied to solve the FCR minimization problem in Section 3. A simulation study of the design is given in Section 4 before presenting the experimental results in Section 5. Finally Section 6 gives the concluding remarks and briefly discusses possible future research.

Notation

Let $u_k[n] \in \mathbb{R}^{N_u}$ denote a signal at trial k and sample n with dimension N_u, and U_k be the super-vector formed from $u_k[n]$ in the finite time interval between the first sample N_s and last sample N_e as

$$U_k = \begin{bmatrix} u_k[N_s]^T & \cdots & u_k[N_e]^T \end{bmatrix}^T \in \mathbb{R}^{N_u N_n} \tag{1}$$

with a total of $N_n = N_e - N_s + 1$ samples. $\tilde{U}_k = u_k[N_e]$ denotes the terminal super-vector. Let a denote a vector then $\|a\| = \sqrt{a^T a}$ denotes the Euclidean norm of this vector and $\|a\|_A = \sqrt{a^T A a}$ denotes the weighted Euclidean norm where A is a positive definite matrix. Let B and C be sets, then $\#B$ denotes the cardinality of B and $B \backslash C = \{x \in B \mid x \notin C\}$ is the difference of B and C.

2 Heuristic Broiler Growth Model and FCR Optimisation

2.1 Heuristic Broiler Growth Model

The heuristic broiler FCR model developed in this section is used to test the data driven broiler growth optimization algorithm developed in Section 3.3 in a simulation environment prior to experimental tests. Only past growth model

data, and not the growth model, is used for control synthesis, which would also be the case under real production conditions. The objective is to represent basic broiler growth behavior in an industrial *state-of-the-art* broiler production, which is based on the experience and knowledge of a broiler application expert.

The model's primary objective is to asses the algorithm's ability to iteratively learn a unique time series of broiler state dependent temperature inputs that minimizes the terminal broiler FCR, while simulating reduced growth for both negatively- and positively suboptimal temperature inputs. Such a broiler growth model can be represented by the discrete time dynamic nonlinear model

$$\begin{bmatrix} x_m[n+1] \\ x_f[n+1] \end{bmatrix} = \begin{bmatrix} x_m[n] \\ x_f[n] \end{bmatrix} + T_s \begin{bmatrix} G(u[n], x_m[n]) \\ R_f(x_m[n]) \end{bmatrix}, \tag{2a}$$

$$\begin{bmatrix} z_w[n] \\ z_f[n] \end{bmatrix} = \begin{bmatrix} R_w(x_m[n]) \\ x_f[n] \end{bmatrix} \text{ and} \tag{2b}$$

$$\begin{bmatrix} y_w[n] \\ y_f[n] \end{bmatrix} = \begin{bmatrix} z_w[n] \\ z_f[n] \end{bmatrix} + \begin{bmatrix} q_w[n] + q_{w,\text{bias}}[n] \\ q_f[n] \end{bmatrix} \tag{2c}$$

with initial conditions $x_m[N_s] = x_f[N_s] = 0$ and measured slaughter weight $\Gamma = z_w[N_e]$, where $x_m[n] \in \mathbb{R}_+$ is the broiler maturity in "effective growth days", $z_w[n] \in \mathbb{R}_+$ is the true broiler weight, $y_w[n] \in \mathbb{R}_+$ is the measured broiler weight, $x_f[n] \in \mathbb{R}_+$ is the cumulative feed consumption, $z_f[n] \in \mathbb{R}_+$ is the true cumulative feed consumption, $y_f[n] \in \mathbb{R}_+$ is the measured cumulative feed consumption, $u[n] \in \mathbb{R}$ is the temperature input, and $T_s \in \mathbb{R}_+$ is the sampling interval in days. Under production conditions the temperature input $u[n]$ is a reference for the climate control system, which for simplicity is assumed to achieve perfect tracking. In (2a), the smooth maturation rate function $G{:}\,\mathbb{R} \times \mathbb{R}_+ \to [\beta, 1]$ where $\beta \in [0, 1[$ is the worst case broiler growth rate. Finally, $R_w{:}\,\mathbb{R}_+ \to \mathbb{R}_+$ and $R_f{:}\,\mathbb{R}_+ \to \mathbb{R}_+$ are smooth and strictly increasing functions mapping the broiler maturity $x_m[n]$ into broiler weight and feed consumption, $q_w[n] \in \mathbb{R}$ is the weight measurement noise, $q_{w,\text{bias}}[n] \in \mathbb{R}$ is the weight bias, and $q_f[n] \in \mathbb{R}$ is the feed measurement noise.

The growth and feed consumption of the widely used ROSS 308 fast growing broiler strain are described by the manufacturer in [Aviagen, 2014, pp. 3] as

$$R_w(t) = \frac{-18.3\,t^3 + 2.2551\,t^2 + 2.9118\,t + 54.739}{1000} \tag{3a}$$

$$R_f(t) = \frac{21.9 \cdot 10^{-6}\,t^4 - 4.232 \cdot 10^{-3}\,t^3 + 0.206\,t^2 + 2.02\,t + 11.6}{1000} \tag{3b}$$

where $R_w(t) \in \mathbb{R}_+$ is the broiler weight reference in kg, $R_f(t) \in \mathbb{R}_+$ is the broiler feed uptake reference in kg/day, and $t \in [0, 59]$ days is the time in "effective growth days". Expressing broiler weight $R_w(x_m[n])$ and broiler feed uptake $R_f(x_m[n])$ in terms of the broiler maturity in "effective growth days" through $x_m[n]$ results in realistic weight and feed uptake behavior, as it accurately captures the nonlinear nature of broiler growth.

A modified normal distribution is an appropriate representation for the maturation rate function G, as it has a unique maximum, and the standard deviation can easily be tuned to design how sensitive G is to temperature errors. Specifically

$$G(u[n],\, x_m[n]) = \beta + (1-\beta)\exp\left\{\ln\left(\frac{\alpha+\beta-1}{\beta-1}\right)\left[\frac{u[n]-\bar{u}(x_m[n])}{\sigma_u}\right]^2\right\}, \quad (4)$$

where $\bar{u}(x_m[n])$ is the temperature maximizing G, $G(\bar{u}(x_m[n]), x_m[n]) = 1$, and $\sigma_u \in \mathbb{R}_+$ is the constant temperature sensitivity. The temperature sensitivity is the temperature input error, $u[n] - \bar{u}(x_m[n])$, resulting in a decreased maturation rate of α – corresponding to $G(\bar{u}(x_m[n]) \pm \sigma_u, x_m[n]) = 1-\alpha$ with $\alpha \in]0, 1-\beta[$. The intention is to allow the broiler application expert to heuristically specify the decreased growth rate for a specific temperature error. In Figure 1, the components of the maturation rate function G are shown. For a more accurate temperature sensitivity, the broilers' feathering and ability to regulate their own body temperature should also be considered, which would possibly make σ_u time and state dependent, but this is left as a subject for future work.

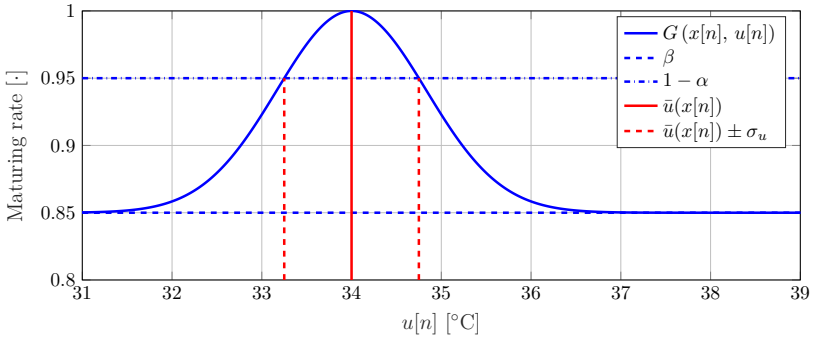


Figure 1: Visualization of the maturation rate function $G(x_m[n], u[n])$ for $x_m[n] = 0$ with worst case broiler growth rate $\beta = 0.85$, $\alpha = 0.05$, maximizing input $\bar{u}(x_m[n]) = 34\,[°\mathrm{C}]$ and temperature error sensitivity $\sigma_u = 0.75\,[°\mathrm{C}]$.

The optimal temperature profile is unknown in the industry, but typical temperature profiles for the ROSS 308 fast growing broiler transition almost linearly between the initial temperature of $\bar{u}_s = 34$ °C at day $t_s = 0$ to $\bar{u}_e = 21$ °C at day $t_e = 34$. This corresponds to a temperature drop of $(\bar{u}_e - \bar{u}_s)$, which is modeled as proportional to the maturity $x_m[n]$ as

$$\bar{u}(x_m[n]) = \bar{u}_s + \Delta T x_m[n] \text{ with } \Delta T = \frac{\bar{u}_e - \bar{u}_s}{t_e - t_s}. \quad (5)$$

Consequently, the optimal temperature at sample n depends on $x_m[n-1]$, which, in turn, depends on all prior inputs.

The weight bias term $q_{w,\text{bias}}[n]$ was investigated in [Johansen et al., 2019a] and found to cause terminal weight measurement errors, $\tilde{y}_w - \tilde{z}_w$, with −27.4g mean and 115.9g standard deviation through comparison with the accurately measured slaughter weight. The weight bias onset was found to occur around day 15, which is heuristically assumed to increase linearly from zero at day 15 to $\mathcal{Q}_{\text{bias}} \sim \mathcal{N}(-27.4\text{g}, 115.9\text{g})$ at the terminal sample and hence

$$z_{w,\text{bias}}[n] = \begin{cases} \frac{nT_s-15}{N_eT_s-15}\mathcal{Q}_{\text{bias}}, & 15 < nT_s \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

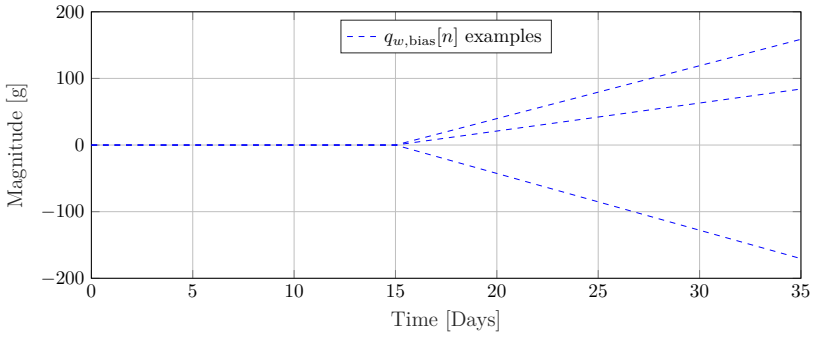
where $\mathcal{Q}_{\text{bias}}$ is constant throughout each simulation as shown in Figure 2a. In [Johansen et al., 2019a] it was found that using the measured slaughter weight, i.e. the terminal broiler weight, reduces the weight bias effect for broiler weight prediction on real broiler production data.

The noise terms $q_w[n]$ and $q_f[n]$ are found by analyzing the frequency spectrum of production data from the experimental test site. As broiler weight is a smooth function of time, the "true" broiler weight is approximated by a second order polynomial $\hat{y}_{w,\text{pol},2}$ between day 3 and 15, where the weight measurement y_w is expected to be the most reliable. The fit errors, $y_w - \hat{y}_{w,\text{pol},2}$, of 36 batches from the experimental test site are shown in the top plot of Figure 2b and is assumed to be measurement noise.

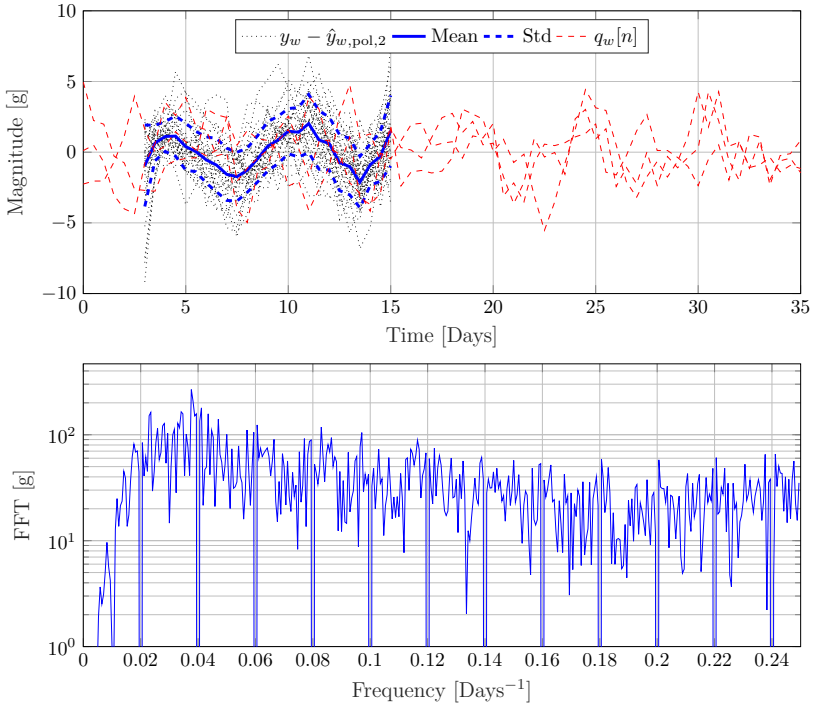
Subtracting the mean, concatenating all the fit errors and computing the FFT produces the bottom magnitude plot. As this is not a standard distribution, random realizations of $q_w[n]$ with identical magnitude are obtained by randomly rotating the phases of the FFT and applying the inverse discrete Fourier transform. For more information on this approach, see [Prichard and Theiler, 1994]. Some realizations of $q_w[n]$ are shown in the top plot of Figure 2b. Similarly, the "true" cumulative feed uptake is approximated by a fourth order polynomial $\hat{y}_{f,\text{pol},4}$ between day 3 and 30 and shown in Figure 2c.

2.2 Control Design Considerations

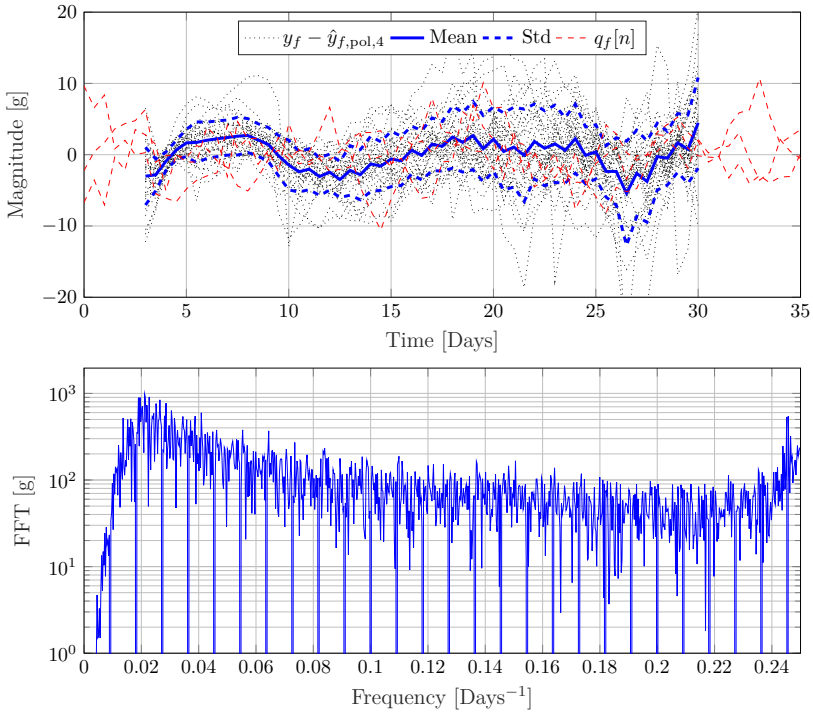
In this section optimization strategies are discussed, where many of these are used in *state-of-the-art* broiler production. Moreover, a heuristic model can explain the rationale between two of the popular strategies considered next. The first of these maintains a higher than necessary temperature to dry out the litter to limit ammonia development and subsequently sickness. The second initially reduce maturation through special light programs to properly develop the broiler, followed by unrestricted growth period for the remaining duration of the batch.



(a) Weight measurement bias $q_{w,\text{bias}}[n]$ samples using (6).



(b) Visualization of the weight measurement noise $q_w[n]$



(c) Visualization of the feed uptake measurement noise $q_f[n]$

Figure 2: Measurement behavior for the heuristic broiler growth model.

Weight maximization considerations

Inspecting G shows that $x_m[n]$ is maximized by the unique input $\bar{u}(x_m[n])$ that for all $u[n] \neq \bar{u}(x_m[n])$ satisfies

$$G(u[n], x_m[n]) < G(\bar{u}(x_m[n]), x_m[n]) = 1.$$

In the case when $\beta \leq G \leq 1$, the largest possible maturity $\bar{x}_m[n]$ equals

$$\begin{aligned}\bar{x}_m[n] &= \max\{x_m[n]\} = T_s \sum_{i=1}^{n} \max\{G(u[i], x_m[i])\} \\ &= n T_s.\end{aligned}$$

As R_w is strictly increasing, the largest possible true broiler weight $\bar{z}[n]$ is given by

$$\begin{aligned}\bar{z}_w[n] &= \max\{z_w[n]\} = \max\{R_w(x_m[n])\} \\ &= R_w(\max\{x_m[n]\}) = R_w(n T_s).\end{aligned}$$

This ensures that suboptimal control results in suboptimal weight, as expected in real broiler production where either a too low or too high temperature results in decreased broiler growth.

In Figure 3 the behavior of the broiler model is shown for different temperature inputs. Interestingly, if the temperature is positively suboptimal, e.g., $u[n] = \bar{u}(\bar{x}_m[n]) + 1\,^\circ\mathrm{C}$, then G converges to 1 for $n \to \infty$, and if the temperature is negatively suboptimal, e.g., $u[n] = \bar{u}(\bar{x}_m[n]) - 1\,^\circ\mathrm{C}$, then G converges to β for $n \to \infty$. This is caused by the decreasing $\bar{u}(x_m[n])$ for increasing $x_m[n]$ – supported by the fact that broiler farmers tend to use positively suboptimal temperatures, similar to the first *state-of-the-art* strategy.

Feed minimization considerations

If $\beta \leq G \leq 1$, the smallest maturation rate $\underline{x}_m[n]$ is governed by

$$\begin{aligned}\underline{x}_m[n] &= \min\{x_m[n]\} = T_s \sum_{i=1}^{n} \min\{G(u[i], x_m[i])\} \\ &= T_s \beta n \end{aligned} \tag{7}$$

As R_f is strictly increasing, the lowest cumulative feed consumption is given by

$$\begin{aligned}\underline{x}_f[n] &= \min\{x_f[n]\} = \min\left\{T_s \sum_{i=1}^{n} R_f(x_m[i])\right\} \\ &= T_s \sum_{i=1}^{n} R_f(\min\{x_m[i]\}) = T_s \sum_{i=1}^{n} R_f(T_s \beta i)\end{aligned} \tag{8}$$

This suggests that feed minimization and weight maximization are completely opposing goals.

FCR minimization considerations

The expression for FCR from the heuristic model is

$$z_{\text{FCR}}[n] = \frac{z_f[n]}{z_w[n]} = \frac{x_f[n]}{R_w(x_m[n])} = T_s \frac{\sum_{i=1}^{n} R_f(x_m[i])}{R_w(x_m[n])} \tag{9}$$

Unlike weight maximization and feed minimization, an analytical expression for the lowest possible FCR is non-trivial to determine, as depicted on Figure 5. This is due to two simultaneous and opposing objectives, namely weight maximization and feed minimization, which turns out to depend on the simulation duration N_e as depicted on Figure 5. In Figure 4 the proposed strategies are compared, from which it follows that FCR minimization consists of an initial period of feed minimization followed by weight maximization – similar to the second *state-of-the-art* strategy. *Feed minimization* produces the highest FCR, and is therefore ruled out. Moreover, *weight maximization* results in a 1.1% higher FCR than *FCR minimization*, which makes *FCR minimization* favorable despite the added complexity of another output, and will therefore be used in this work.

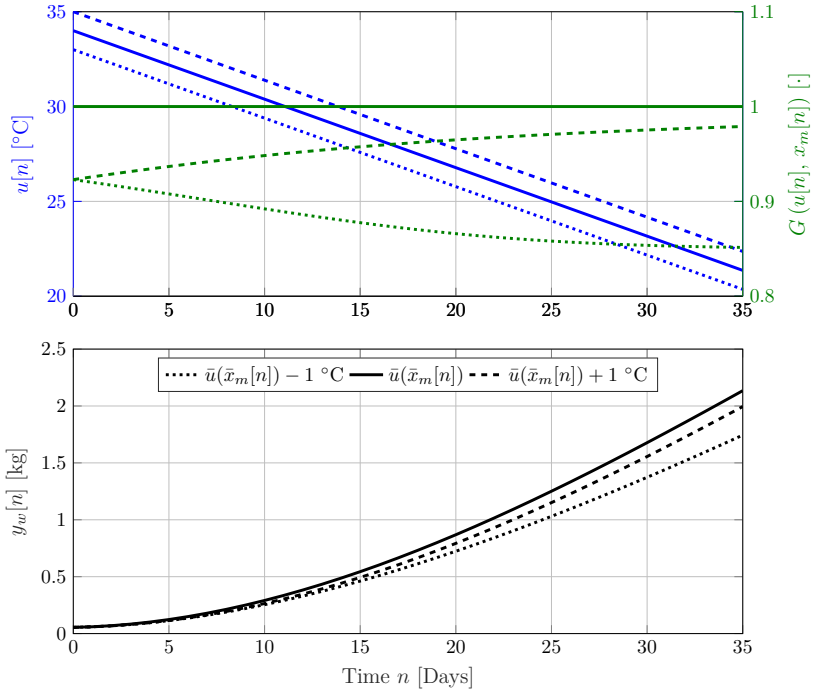


Figure 3: Visualization of broiler growth y_m with different inputs. The top plot depicts the maturation rate function $G(x_m[n], u[n])$ as a function of the input $u[n]$ and the bottom plot depicts the output $y_m[n]$. The model settings equal that of Figure 1 with $T_s = 1$ day.

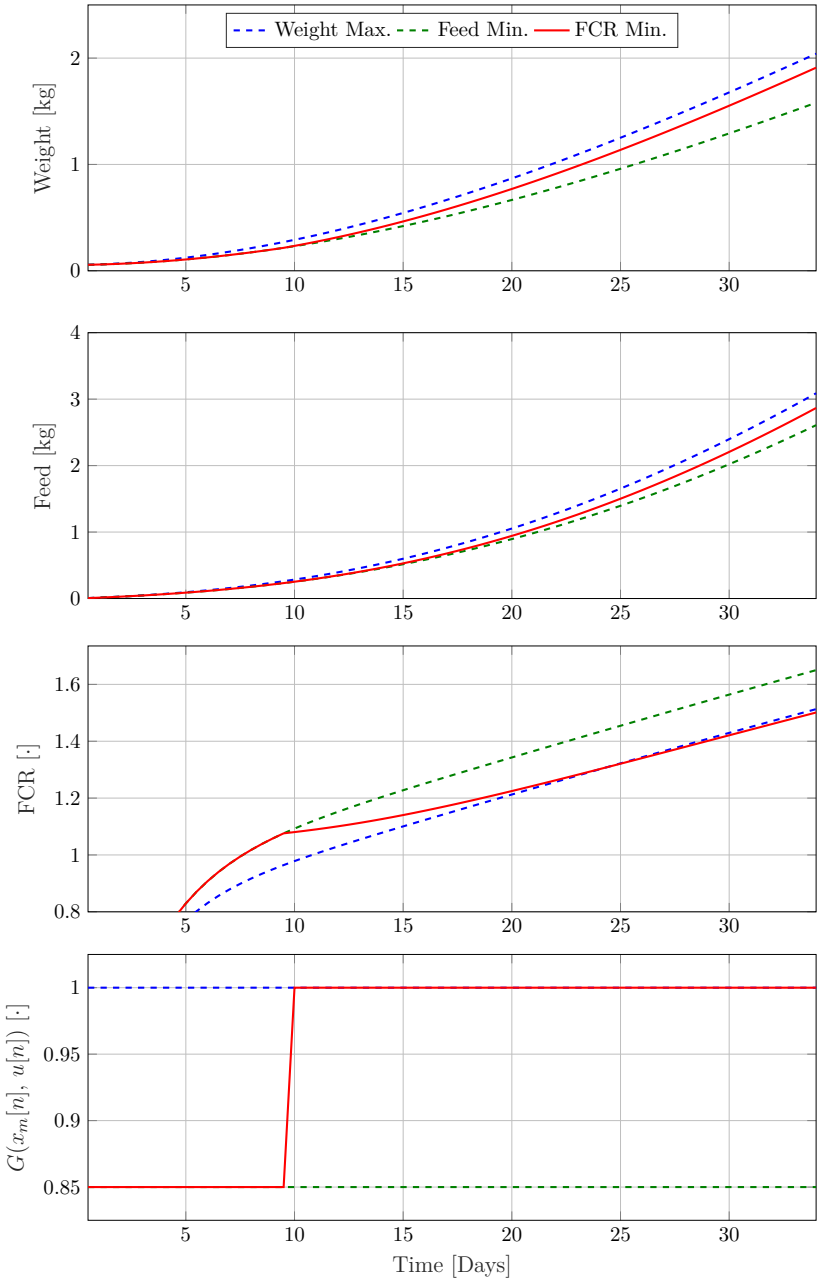


Figure 4: Visualization of different optimization strategies with N_e = 34 days, T_s = 1 day and β = 0.85. A FCR difference of $1.14 \cdot 10^{-3}$, equivalent of 1.1%, exists between *Growth maximization* and *FCR minimization*, which potentially makes *FCR minimization* a better strategy.

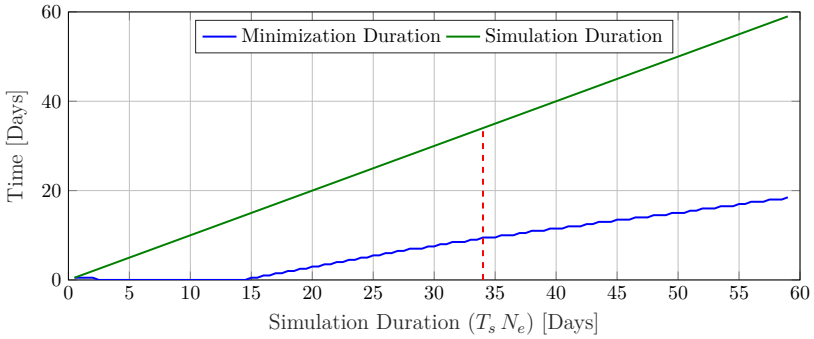


Figure 5: Minimization duration as a function of simulation duration ($N_e T_s$) with $\beta = 0.85$ and $T_s = 0.5$ days, which corresponds to the length of the initial period where $G(x_m[n], u[n]) = \beta$. The red dashed line indicates a simulation duration of 34 days with a minimization duration of 9.5 days, equivalent to Figure 4

3 Broiler FCR Minimization using Terminal ILC

3.1 Terminal Iterative Learning Control (TILC)

TILC is a method that can be applied to a repeating process with the aim of iteratively learning the input sequence $U_k \in \mathbb{R}^{N_u N_n}$ such that the terminal process output $\tilde{Y}_k(U_k) \in \mathbb{R}^{N_y}$ tracks the desired terminal reference $\tilde{R} \in \mathbb{R}^{N_y}$ denoted by

$$\lim_{k\to\infty} \tilde{Y}_k(U_k) = \tilde{R}, \tag{10}$$

with the super-vector model used for control synthesis given by

$$\tilde{Y}_k(U_k) = \tilde{P}U_k + \tilde{K}, \tag{11}$$

where $\tilde{P} \in \mathbb{R}^{N_y \times N_u N_n}$ is the terminal system matrix, and $\tilde{K} \in \mathbb{R}^{N_y}$ represents terminal effects unrelated to the input $U \in \mathbb{R}^{N_u N_n}$.

This last problem can be solved using constrained Norm Optimal Point-To-Point ILC, which aims at tracking the output at specific samples. As TILC only aims at tracking the terminal output, TILC is a specialization of Point-To-Point ILC. Adapting the constrained Norm Optimal Point-To-Point ILC algorithm 1 in [Chu et al., 2015] to the special case to the TILC problem gives

$$U_{k+1} = \underset{U\in\Omega}{\arg\min} \; \|\tilde{E}_k(U)\|^2_{W_{\tilde{E}}} + \|U - U_k\|^2_{W_{\Delta U}} \tag{12a}$$

subject to

$$\tilde{E}_k(U) = \tilde{R} - \tilde{Y}_k(U) \text{ and} \tag{12b}$$

$$\tilde{Y}_k(U) = \tilde{P}U + \tilde{K}, \tag{12c}$$

where Ω is the set of valid inputs, $W_{\tilde{E}} \in \mathbb{R}^{N_y \times N_y}$ is the symmetric positive definite tracking error cost matrix, $W_{\Delta U} \in \mathbb{R}^{N_u N_n \times N_u N_n}$ is the symmetric positive definite input change cost matrix and $\tilde{E}_k(U) \in \mathbb{R}^{N_y}$ is the terminal tracking error given by (12b). The intuition behind (12) is to reduce the terminal tracking error by finding an input in the neighborhood of U_k that minimizes the cost function (12a).

The following results were established in [Chu et al., 2015].

Theorem 3. *If perfect tracking is feasible, i.e.* $\exists U \in \Omega$ *such that* $\tilde{Y}_k(U) = \tilde{R}$*; then (12) achieves monotonic convergence to zero tracking error*

$$\left\|\tilde{E}_{k+1}(U_{k+1})\right\|_{W_{\tilde{E}}} \leq \left\|\tilde{E}_k(U_k)\right\|_{W_{\tilde{E}}} \quad \forall k \in \mathbb{Z}_+ \tag{13}$$

and

$$\lim_{k\to\infty} \tilde{E}_k(U_k) = 0, \lim_{k\to\infty} U_k = \bar{U}. \tag{14}$$

Theorem 4. *If perfect tracking is not feasible, i.e.* $\tilde{Y}_k(U) \neq \tilde{R} \quad \forall U \in \Omega$*; then the input of (12) converges to*

$$\lim_{k\to\infty} U_{k+1} = \arg\min_{U\in\Omega} \|\tilde{R} - \tilde{P}U - \tilde{K}\|^2_{W_{\tilde{E}}}, \tag{15}$$

equivalent of the algorithm converging to the smallest possible tracking error. Moreover, this convergence is monotonic in the tracking error norm

$$\left\|\tilde{E}_{k+1}(U_{k+1})\right\|_{W_{\tilde{E}}} \leq \left\|\tilde{E}_k(U_k)\right\|_{W_{\tilde{E}}} \quad \forall k \in \mathbb{Z}_+. \tag{16}$$

3.2 Data Driven Model

The objective of the data driven model is to enable control synthesis without a mathematical broiler FCR model, by synthesizing $\tilde{P}$ and $\tilde{K}$ from (12c) using past production data. Using a nonlinear discrete time data driven model the aim is to capture the broiler growth dynamic using data from the past N_b trials, $\{\{U_{k-N_b+1}, D_{k-N_b+1}, Y_{k-N_b+1}\}, \ldots, \{U_k, D_k, Y_k\}\}$, where D_k denotes the disturbance vector, where N_b data indexes are conveniently denoted by

$$\mathcal{B}_k = \{k - N_b + 1, \ldots, k\}. \tag{17}$$

For data driven model synthesis at trial k, data from the trial indexes denoted by $\mathcal{B}_{k-1}$ is required. Trial data prior to the first trial, k<1, are denoted as *preliminary* trials, e.g., $\{U_{-2}, D_{-2}, Y_{-2}\}$. Hence, a total of N_b preliminary trials are required for model synthesis for the first trial, $k = 1$, denoted by the indexes $\mathcal{B}_0 = \{1 - N_b, \ldots, 0\}$.

The data driven model is chosen to be a NARMAX type neural network model with N_l input and output lags, a single hidden layer with N_N neurons and hyperbolic tangent activation function in the hidden layer and hence

$$\hat{y}_k[n+1 \mid \mathcal{W}, s] = W^{\mathrm{o}} \tanh\left(\mathcal{X} + \theta^{\mathrm{h}}\right) + \theta^{\mathrm{o}} \tag{18}$$

with

$$\mathcal{X} = \sum_{i=0}^{N_l-1} W^{\mathrm{h}}_{y,i}\hat{y}_k[n-i \mid \mathcal{W}, s] + W^{\mathrm{h}}_{u,i}u_k[n-i] + W^{\mathrm{h}}_{d,i}d_k[n-i],$$

where $W^{\mathrm{o}} \in \mathbb{R}^{N_y \times N_N}$, $\mathcal{X} \in \mathbb{R}^{N_N}$, $\theta^{\mathrm{o}} \in \mathbb{R}^{N_N}$, $W^{\mathrm{h}}_{y,i} \in \mathbb{R}^{N_N \times N_y}$, $W^{\mathrm{h}}_{u,i} \in \mathbb{R}^{N_N \times N_u}$, $W^{\mathrm{h}}_{d,i} \in \mathbb{R}^{N_N \times N_d}$ and $\theta^{\mathrm{h}} \in \mathbb{R}^{N_N}$ are model parameters stored in $\mathcal{W}$, while $\hat{y}_k[n \mid \mathcal{W}, s]$ is the model output at sample n, initialized at sample s with model weights $\mathcal{W}$. Initialization in this case refers to

$$\hat{y}_k[n \mid \mathcal{W}, s] = y_k[n] \quad \forall n \leq s, \tag{19}$$

where n is implicitly lower bounded by the starting sample N_s, $N_s \leq n$, for both $y_k[n]$, $u_k[n]$ and $d_k[n]$. To find the model weights, the following training procedure was used

$$\mathcal{W}(B) = \arg\min_{\mathcal{W}} \sum_{b \in B \setminus \min\{B\}} \frac{J_b(\mathcal{W})}{\#B - 1} \tag{20a}$$

with

$$J_b(\mathcal{W}) = \bar{\alpha}\|\mathcal{W}\|^2 + \sum_{i=1}^{N_S}\sum_{n=S_i}^{N_e} \frac{\mathcal{E}_b}{N_y(N_e - S_i + 1)} \tag{20b}$$

$$\mathcal{E}_b = \sum_{i=1}^{N_y} \begin{cases} \|\Gamma_b - \hat{y}_i\|_2^2\,\phi(k), & k = N_{s,b} \wedge i = i_w \\ \|y_i - \hat{y}_i\|_2^2\,\phi(k), & k \neq N_{s,b} \wedge i = i_w \\ \|y_i - \hat{y}_i\|_2^2, & \text{otherwise} \end{cases} \tag{20c}$$

$$\phi(k) = \begin{cases} 1, & k < N_\phi \\ 1 + (N_{s,b} - N_\phi)(\gamma - 1), & k = N_{s,b} \\ \gamma, & \text{otherwise} \end{cases} \tag{20d}$$

where B is a set of batch-indices used for training, $S = \{S_1, \ldots, S_{N_S}\}$ is the set of $N_S \in \mathbb{Z}_+$ initialization locations, which was found to speed up training as described in [Johansen et al., 2019b], Γ_b is the broiler slaughter weight of batch b, i.e. the true broiler weight prior to slaughter, $i_w \in \mathbb{Z}$ is the weight output index, $\phi : \mathbb{Z}_+ \to \mathbb{R}$ is the weight cost shaping function, $N_\phi \in \mathbb{Z}$ is the start weight cost shaping sample number and $\gamma \in]0, 1[$ is the weight cost shaping parameter.

Automatic weighing pads are commonly used for weighing broilers and is known to be negatively biased onwards from day 15, which is represented by (6) in the heuristic model. In [Johansen et al., 2019a] the weight cost shaping function $\phi : \mathbb{Z}_+ \to \mathbb{R}$ in (20c) and (20d) was found to decrease the impact of this bias – one example of ϕ is shown in Figure 6. The slaughter weight is considered very accurate and is included by overriding the last measured local weight at sample $k = N_{s,b}$ of each batch. Extra emphasis is then placed on the slaughter weight at sample $k = N_{s,b}$ in the cost function, while samples beyond $N_\phi \in \mathbb{Z}_+$ are weighted less.

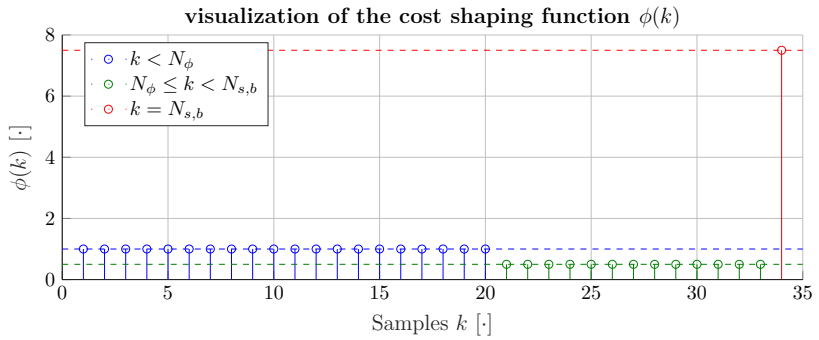


Figure 6: Visualization of the cost shaping function $\phi(k)$ with $N_\phi = 20$, $N_{s,b} = 34$ and $\gamma = 0.5$. The blue, green and red values correspond to a separate case of (20d).

The cost function is minimized using the Levenberg-Marquardt algorithm with early stopping applied on the oldest batch index in B, denoted by $\min\{B\}$, in $J_{\min\{B\}}(\mathcal{W})$, to prevent overtraining. The regularization constant $\bar{\alpha} \in \mathbb{R}_+$ is found iteratively through Bayesian regularization to prevent overfitting. The model weights $\mathcal{W}$ are initialized using the Nguyen Widrow initialization scheme. For detailed information regarding the training see [Johansen et al., 2019b] and [Johansen et al., 2019a].

As (20a) is not a convex optimization problem, the weights $\mathcal{W}(B)$ are not guaranteed to be the global minimum. To decrease the probability of ending up in a local minimum, the ensemble mean of N_m models trained with different initial model weights is used. The ensemble data driven model simulated from sample N_s with data from batch b, $\{Y_b, D_b, U_b\}$, is

$$\hat{y}_{k,b}[n] = \frac{1}{N_m} \sum_{l=1}^{N_m} \hat{y}_b[n \mid \mathcal{W}_l(\mathcal{B}_k \backslash b), N_s], \tag{21}$$

where $\mathcal{W}_l(\mathcal{B}_k \backslash b)$ is the l'th training of $\mathcal{W}(\mathcal{B}_k \backslash b)$ with the batch indexes $\mathcal{B}_k \backslash b$ to separate training data and simulation data. The terminal super-vector ensemble data driven model required for (12) is obtained by linearizing (21) along the trajectory of U_b using the first order Taylor expansion

$$\tilde{Y}_k(U) \approx \hat{\hat{Y}}_{k,b} + \hat{\hat{P}}_{k,b}(U - U_b) = \hat{\hat{P}}_{k,b} U + \hat{\hat{K}}_{k,b} \tag{22}$$

with

$$\hat{\hat{P}}_{k,b} = \left. \frac{d\hat{\hat{Y}}_{k,b}}{dU_b^T} \right|_{U_b} \text{ and } \hat{\hat{K}}_{k,b} = \hat{\hat{Y}}_{k,b}(U_b) - \hat{\hat{P}}_{k,b} U_b$$

where $U \in \mathbb{R}^{N_u N_n}$ is the super-vector input used in (12c). The data driven model is retrained for every k and b.

To use this model for FCR minimization requires an augmented data driven model, denoted by $(\cdot)^*$. This model is given by

$$\tilde{Y}_k^*(U) = \frac{\tilde{Y}_{k,f}(U)}{\tilde{Y}_{k,w}(U)} \tag{23}$$

where $\tilde{Y}_{k,w}(U) \in \mathbb{R}_+$ and $\tilde{Y}_{k,f}(U) \in \mathbb{R}_+$ respectively denote the weight and cumulative feed uptake – the equivalent of (2c). Linearized in U_b by a first order Taylor expression similar to (22) results in

$$\tilde{Y}_k^*(U) \approx \hat{\tilde{Y}}_{k,b}^*(U_k) + \hat{\tilde{P}}_{k,b}^*(U - U_k) = \hat{\tilde{P}}_{k,b}^* U + K_{k,b}^* \tag{24}$$

with

$$\hat{\tilde{P}}_{k,b}^* = \frac{d\hat{\tilde{Y}}_{k,b}^*(U)}{d\hat{\tilde{Y}}_{k,b}^T(U)} \frac{d\hat{\tilde{Y}}_{k,b}(U)}{dU^T} = \frac{d\hat{\tilde{Y}}_{k,b}^*(U)}{d\hat{\tilde{Y}}_{k,b}^T(U)} \hat{\tilde{P}}_{k,b} \text{ and}$$

$$\hat{\tilde{K}}_{k,b}^* = \hat{\tilde{Y}}_{k,b}^*(U_k) - \hat{\tilde{P}}_{k,b}^* U_k = \frac{d\hat{\tilde{Y}}_{k,b}^*(U)}{d\hat{\tilde{Y}}_{k,b}^T(U)} \hat{\tilde{K}}_{k,b}.$$

3.3 Data Driven TILC Broiler FCR Minimization

The objective is to minimize terminal broiler FCR, which is unknown in broiler production. One reason for this is that artificial genetic selection progressively increases the growth rate. To account for this, the reference is redefined to

$$\tilde{R}_k^* = \tilde{Y}_k^*(U_k) - \mathcal{R}, \tag{25}$$

where $\mathcal{R} \in \mathbb{R}_+^{N_y}$ is a trial-independent minimization vector with positive elements and this method is termed *minimizing reference*. As $\tilde{E}_k^*(U_k) = \tilde{R}_k^* - \tilde{Y}_k^*(U_k) = -\mathcal{R}$ is constant, zero tracking error is not possible by construction. Assuming that $\tilde{Y}_k^*(U_k)$ is lower bounded by $\tilde{Y}_{\min}^* \in \mathbb{R}^{N_y}$ and in combination with Theorem 4, the aim is to achieve

$$\lim_{k\to\infty} \tilde{Y}_k^*(U_k) = \tilde{Y}_{\min}^* \text{ and } \lim_{k\to\infty} \tilde{R}_k^* = \tilde{Y}_{\min}^* - \mathcal{R}. \tag{26}$$

Since broiler growth is a nonlinear process, a local minimum could be obtained instead of $\tilde{Y}_{\min}^*$.

In the following the so-called best recent trial index κ_k is required and for $\tilde{Y}_i^*(U_i) \in \mathbb{R}_+$ it is defined by

$$\kappa_k = \underset{i\in[\min(k-N_b,0),\,k]}{\arg\min} \|\tilde{Y}_i^*(U_i)\|_{W_{\tilde{E}}}, \tag{27}$$

and serves as a feasible substitute for the *true best recent trial index* given by

$$\underset{i\in[\min(k-N_b,0),\,k]}{\arg\min} \|\tilde{Y}_{\min}^* - \tilde{Y}_i^*(U_i)\|_{W_{\tilde{E}}}.$$

The variable i is lower bounded by 0, which equals the most recent preliminary trial and (27) is application-dependent. To reduce the influence of the measurement weight bias on κ_k, the slaughter weight Γ_k and measured cumulative feed consumption $\tilde{Y}_{k,f}(U_k)$ is used:

$$\kappa_k = \underset{i\in[\min(k-N_b,0),k]}{\arg\min} \left\| \frac{\tilde{Y}_{i,f}(U_i)}{\Gamma_i} \right\|_{W_{\tilde{E}}}. \tag{28}$$

To account for the uncertain nature of the augmented data driven model given by (24), the TILC algorithm is modified into a descent type algorithm, denoted *anchoring*, by solving

$$U_{k+1} = \underset{U\in\Omega_{k+1}}{\arg\min} \left\|\tilde{E}^*_{\kappa_k}(U)\right\|^2_{W_{\tilde{E}}} + \|U - U_{\kappa_k}\|^2_{W_{\Delta U}} \tag{29a}$$

subject to (25), (28) and

$$\tilde{E}^*_{\kappa_k}(U) = \tilde{R}^*_{\kappa_k} - \tilde{Y}^*_{\kappa_k}(U) \text{ and} \tag{29b}$$

$$\tilde{Y}^*_{\kappa_k}(U) = \hat{\tilde{P}}^*_{k,\kappa_k} U + \hat{\tilde{K}}^*_{k,\kappa_k} \tag{29c}$$

where $\Omega_{k+1} \in \mathbb{R}^{N_u N_n}$ is the set of valid trial dependent inputs.

If the input U_{k+1} does not decrease the error in (28), then the data driven model is not sufficiently accurate in the neighborhood of U_{κ_k}, i.e. $\hat{\tilde{P}}^*_{k,\kappa_k} \neq \tilde{P}^*_k$. Therefore U_{k+1} is rejected and U_{k+2} is calculated in the neighborhood of $U_{\kappa_{k+1}} = U_{\kappa_k}$ instead of U_{k+1}. This effectively ensures that the algorithm keeps exploring the neighborhood of the recent best trial input U_{κ_k} until the data driven model is sufficiently accurate to maximize the terminal output norm in (28), as the data driven model always uses the most recent data from the last N_b trials. Consequently, if the data driven model $\hat{\tilde{P}}^*_{k,\kappa_k}$ is identical to the analytical model $\tilde{P}^*_{\kappa_k}$ under ideal conditions and constant reference. In this case $\kappa_k = k$ as $\tilde{E}^*_k$ is monotonically decreasing in k.

The computable solution of (29) is

$$U_{k+1} = U_{\kappa_k} + \underset{\Delta U\in\Omega_{k+1}-U_{\kappa_k}}{\arg\min} \frac{1}{2}\|\Delta U\|^2_{Q_1} + Q_2^T \Delta U \tag{30}$$

where

$$Q_1 = 2\left(\hat{\tilde{P}}^{*T}_{k,\kappa_k} W_{\tilde{E}} \hat{\tilde{P}}^*_{k,\kappa_k} + W_{\Delta U}\right) \text{ and}$$
$$Q_2 = -2\hat{\tilde{P}}^{*T}_{k,\kappa_k} W_{\tilde{E}} \tilde{E}^*_{\kappa_k}(U_{\kappa_k})$$

and $\Delta U = U - U_{\kappa_k}$ results in an algorithm of the form $U_{k+1} = F\left(U_{\kappa_k}, \tilde{E}^*_{\kappa_k}(U_{\kappa_k})\right) = F\left(U_{\kappa_k}, \tilde{R}^*_{\kappa_k} - \tilde{Y}^*_{\kappa_k}(U_{\kappa_k})\right)$ that includes feedback action though the measured terminal output via the terms $\tilde{Y}^*_{\kappa_k}(U_{\kappa_k})$ and $\hat{\tilde{P}}^*_{k,\kappa_k}$. The slaughter weight is used

to calculate $\tilde{E}^*_{\kappa_k}(U_{\kappa_k})$, similar to (28), to reduce the influence of the weight measurement bias. If combined with maximizing reference then $\tilde{E}^*_k(U_{\kappa_k}) = \mathcal{R}$ and $\tilde{Y}^*_{\kappa_k}(U_{\kappa_k})$ is only used indirectly through $\hat{\tilde{P}}^*_{k,\kappa_k}$. This problem can be solved using standard quadratic programming solvers, e.g. Matlab's `quadprog`.

3.4 Analytical Heuristic Model

To evaluate the ILC algorithm formulated in Section 3.3 in simulation, an analytical linear terminal super-vector broiler growth model of $\tilde{Z}_k \in \mathbb{R}^{N_y}$ is required. This is obtained by linearizing (2) along the trajectory of $U_k \in \mathbb{R}^{N_u N_n}$ using the first order Taylor expansion:

$$\tilde{Z}_k(U) \approx \tilde{Z}_k(U_k) + \tilde{P}_k (U - U_k) = \tilde{P}_k U + \tilde{K}_k \tag{31}$$

with

$$\tilde{P}_k = \left.\frac{d\,\tilde{Z}_k(U_k)}{d\,U_k^T}\right|_{U_k} \text{ and } \tilde{K}_k = \tilde{Z}_k(U_k) - \tilde{P}_k U_k,$$

where $\tilde{P}_k \in \mathbb{R}^{N_y \times N_u N_n}$ is the terminal model matrix and $\tilde{K} \in \mathbb{R}^{N_y}$ is the terminal output constant vector unrelated to the input $U \in \mathbb{R}^{N_u N_n}$.

4 Simulation Case Study

4.1 Description

The objective is to investigate the ability of different configurations of the data driven optimization algorithm (29) to minimize the terminal FCR $\tilde{Y}^*_k$ of the heuristic broiler growth model given by (2). Specifically, the performance impact of the following is investigated:

1. using the data driven model $\hat{\tilde{P}}^*_{k,\kappa_k}$ for control synthesis from (22), denoted by (D), compared to unrealistic option of using the analytical super-vector model $\tilde{P}^*_{\kappa_k}$ for control synthesis from (31), denoted by (I), as shown in Figure 7b.

2. using anchoring from (29) though κ_k from (28), denoted by (A), compared to disabling this term by forcing $\kappa_k = k$, denoted by $(\cdot)$, as shown on Figure 7c.

3. using the maximizing reference (25), denoted by (MR), compared to unrealistic option of using the true analytic maximum given by

$$\tilde{R}^*_k = \tilde{Y}_{\min} = \underline{z}[N_e], \tag{32}$$

indicated by $(\cdot)$, as depicted on Figure 7d.

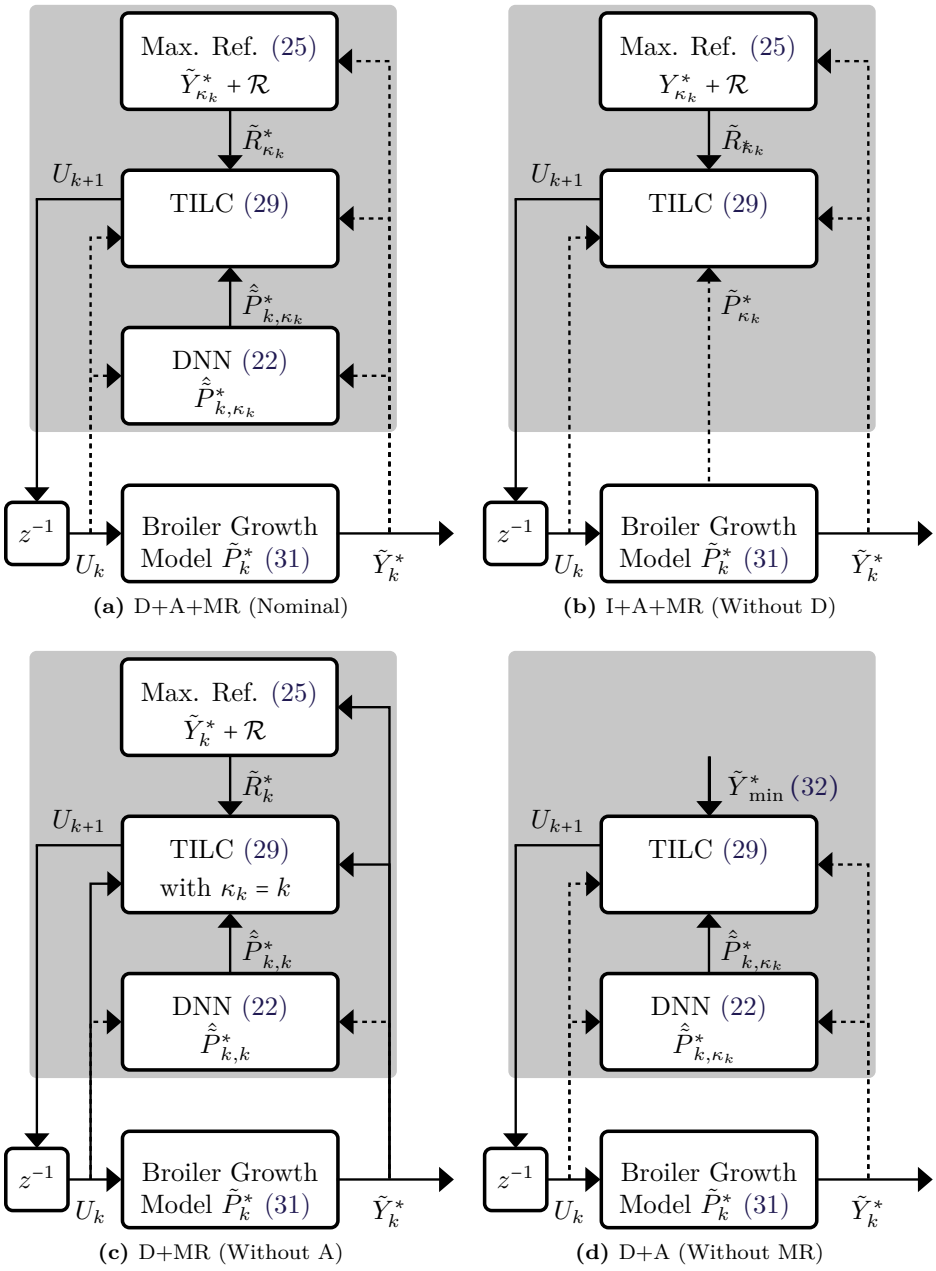


Figure 7: Illustration of some of the configurations of the broiler growth optimization algorithm tested in Section 4. The shaded area denotes the controller, z^{-1} denotes a unit delay, a dashed signal contains information from the last N_b trials, $\{k - N_b + 1, \ldots, k\}$, and a non-dashed signal only contains information from trial k. See Section 4.1 for detailed explanation.

This results in a total of 8 different test configurations, some of which are shown in Figure 7. Each test is repeated 10 times and the mean true terminal error, $|\tilde{Z}_k - \tilde{R}_{\max}|$, is used for evaluation.

To investigate the necessity for iterative learning in this data driven application, different values of $W_{\Delta U}$ are explored under unconstrained conditions, i.e., $\Omega_k = \mathbb{R}^{N_e N_u}$, e.g. by using $W_{\Delta U} = 0$ with a perfect model under linear conditions results in instantaneous convergence in a single trial [sung Ahn et al., 2006]. Specifically, if $W_{\Delta U} = 0$ has instantaneous convergence with the D+A+MR algorithm compared to using $W_{\Delta U} > 0$, then there is no need for iterative learning.

4.2 Method and Model Configuration

The heuristic broiler growth model in Section 2 was simulated between the initial sample $N_s = 0$ and the terminal sample $N_e = 35$ with a sample interval of $T_s = 0.5$ days, and is heuristically configured as follows: $\beta = 0.85$ as the worst case maturing rate, since feed and water consumption are the dominating factors and correct temperature control is regarded as a catalyst. Also $\alpha = 0.05$ and $\sigma_u = 0.75\,[^\circ\mathrm{C}]$ have been used to give good overall sensitivity throughout the lifespan of a broiler.

The data driven model in Section 3.2 is generated with $N_m = 20$ ensemble models using $N_b = 10$ preliminary training batches, $N_l = 3$ input and output lags, $N_N = 7$ neurons in the hidden layer and with $N_S = 5$ initialization locations at samples $S = \{0, 7, 14, 21, 28\}$. The preliminary N_b trials required for training are generated using the positive input $u[n]$ resulting in a 5% decreased maturing rate, $G(u[n], x_m[n]) = 0.95$, see the example in Figure 8. To ensure an identical initial input U_0 for all the tests, the most recent preliminary trial $k = 0$ does not have any added input noise. Hence, the objective is to decrease the terminal broiler FCR $\tilde{Y}_k^\star$ by 0.0537. White noise with standard deviation of 0.3 °C is added to the remaining $N_b - 1$ preliminary trials, $\{1 - N_b, \ldots, -1\}$. This is considered realistic, as most broiler farmers tend to use a too high temperature with little variations from trial-to-trial.

Fast convergence conditions for the data driven TILC broiler optimization algorithm are obtained by using a minimization constant of $\mathcal{R} = 0.04$, terminal tracking error cost and input change cost of $W_{\tilde{E}} = 0.01^{-2}$ and $W_{\Delta U} = \mathrm{diag}\big([1\,^\circ\mathrm{C}]^{-2}, \ldots, [1\,^\circ\mathrm{C}]^{-2}\big)$. The permitted temperature change is restricted to avoid large input fluctuations caused by data driven modeling errors in $\hat{\tilde{P}}^\star_{k,\kappa_k}$. The valid input space Ω_{k+1} is therefore given by:

$$\omega_{k+1}[n] = \{u \,|\, -\gamma[n] \le u - u_{\kappa_k}[n] \le \gamma[n]\} \text{ with} \tag{33}$$
$$\gamma[n] = 0.5\,^\circ\mathrm{C} + n\,T_s\,\frac{1.5\,^\circ\mathrm{C}}{35\,\mathrm{Days}}$$

where $u \in \mathbb{R}$ is the input and $\gamma[n]$ is the lower and upper temperature change bound ranging from 0.5 °C on day 0 to 2 °C on day 35. This does not restrict the permitted input space Ω_{k+1} for $k \to \infty$ as it changes with $u_{\kappa_k}[n]$.

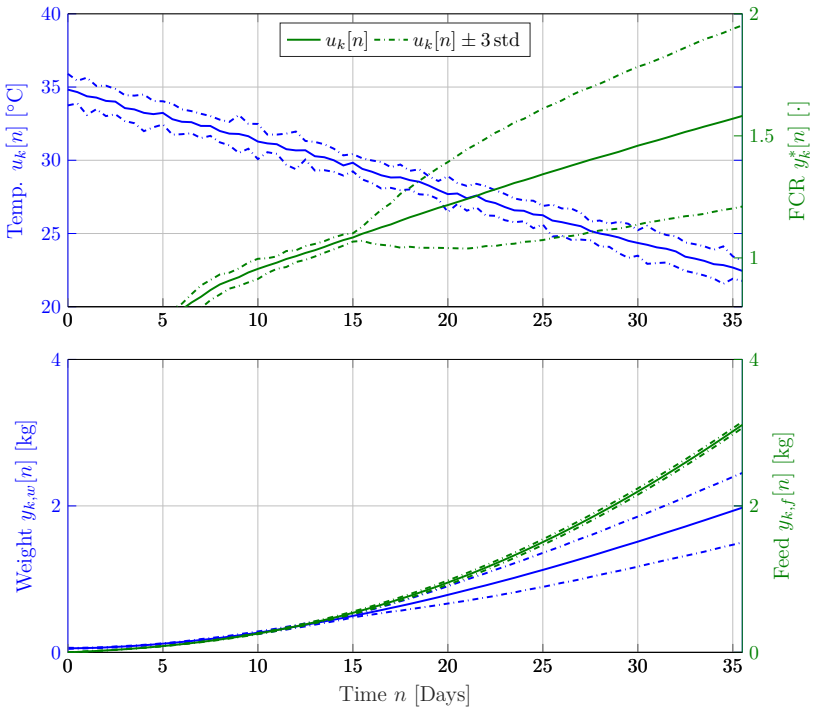


Figure 8: Visualization of 10 preliminary trial data. Note that the large FCR standard deviation is caused by by the measured weight bias $q_{w,\text{bias}}[n]$.

4.3 Results

From Figure 9a it can be concluded that anchoring does not provide benefits under ideal modeling conditions, as I and I+A are almost identical – exactly as expected. However, anchoring is beneficial in conjunction with the data driven model, as D fails at minimizing FCR while D+A converges, but significantly slower than, e.g., I. This makes anchoring superior under data driven modeling conditions.

From I+MR on Figure 9b it can be concluded that using maximizing reference produces similar results to the unrealistic case where the smallest possible FCR is known. Also MR does not improve convergence conditions with a data driven model, since D+MR and D do not converge to zero error.

Using both MR and A, as shown in Figure 9c, leads to the conclusion that D+MR+A is the best performing implementable configuration of the algorithm, as D does not converge despite I and I+MR+A having superior performance. The convergence difference between I and D+MR+A is significant and is most notably caused by the measured weight bias $q_{w,\text{bias}}[n]$. To demonstrate that this is the case, removing the bias results in Figure 9d by enforcing $q_{w,\text{bias}}[n] = 0$, which has a slightly slower convergence rate compared to I and

also a final FCR offset of ≈ 0.01.

In Figure 9e the D+MR+A algorithm is shown with different input change cost $W_{\Delta U}$, which demonstrates that if $W_{\Delta U}$ is configured too low, then the algorithm does not converge. Moreover, it suggests that iterative learning is required to solve the data driven FCR minimization problem and that TILC provides one possible solution.

5 Experimental Study

The results in this section are from an experimental study undertaken in a 20 year old *state-of-the-art* broiler house situated in northern Denmark, also considered in [Johansen et al., 2019b] and [Johansen et al., 2019a]. Each batch approximately contains 40,000 ROSS 308 broilers and an average duration of 34 days. A single trial conducted between June 27 and August 30, 2018, is detailed in the following.

5.1 Method Modification

This section details the modifications necessary for experimental testing of the D+A+MR algorithm developed in Section 3.3.

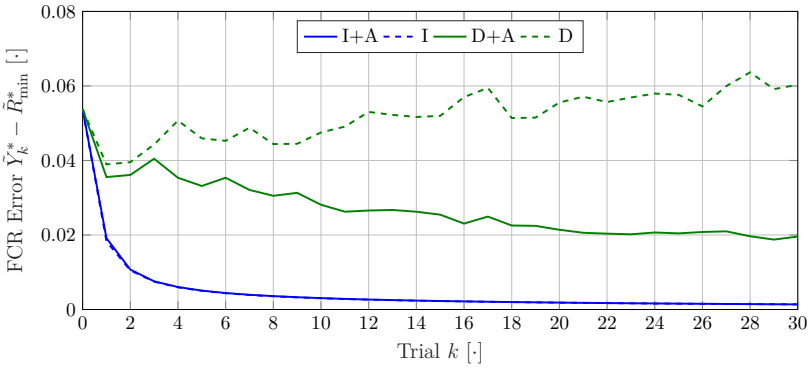
Input Variable Selection (IVS)

For detailed information about the IVS algorithm see [Johansen et al., 2019b]. *State-of-the-art* broiler production typically execute 5-8 batches per house per year. The production parameters naturally change over time as the broiler house deteriorates and both the broiler and feed performance increases. This effectively results in a parameter-drift, which drastically reduces the amount of usable production data (although the parameter-drift rate has not yet been investigated). Furthermore, data quantity requirement scales exponentially with the number of inputs, input lags and output lags for the algorithm[Johansen et al., 2019b]. To alleviate this problem, mutual information based IVS is used to select the most significant inputs, input lags and output lags to make best use of the available production data.

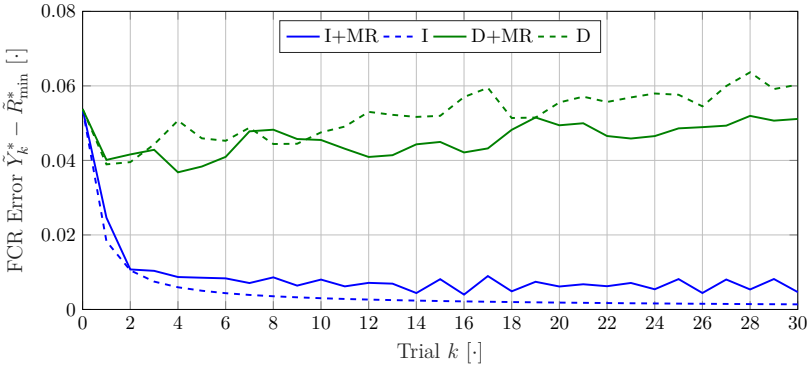
The IVS result is included by modifying the structure of $W^{\mathrm{h}}_{u,i}$, $W^{\mathrm{h}}_{y,i}$, and $W^{\mathrm{h}}_{d,i}$. For example, if the disturbances indexed 1 and 3 are selected with delay of $i = 2$, $N_d = 4$ disturbances, $N_h = 3$ hidden neurons, then $W^{\mathrm{h}}_{d,i}$ is

$$W^{\mathrm{h}}_{d,2} = \begin{bmatrix} \mathcal{W}_1 & 0 & \mathcal{W}_2 & 0 \\ \mathcal{W}_3 & 0 & \mathcal{W}_4 & 0 \\ \mathcal{W}_5 & 0 & \mathcal{W}_6 & 0 \end{bmatrix}. \tag{34}$$

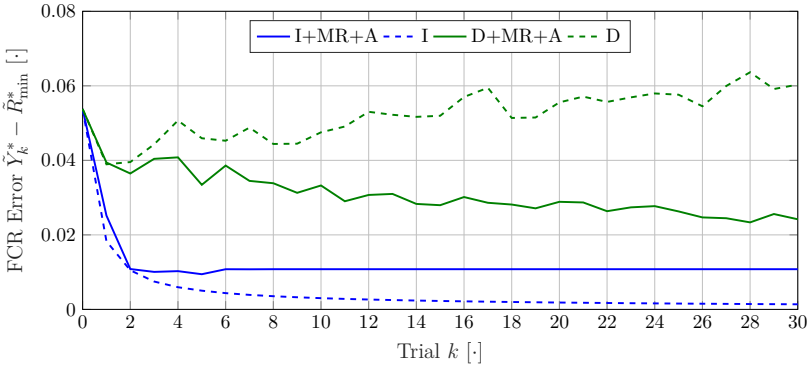
All inputs and outputs are not guaranteed to be present in all the available batches. To maximize the amount of available information, up to $N_b \in \mathbb{Z}_+$



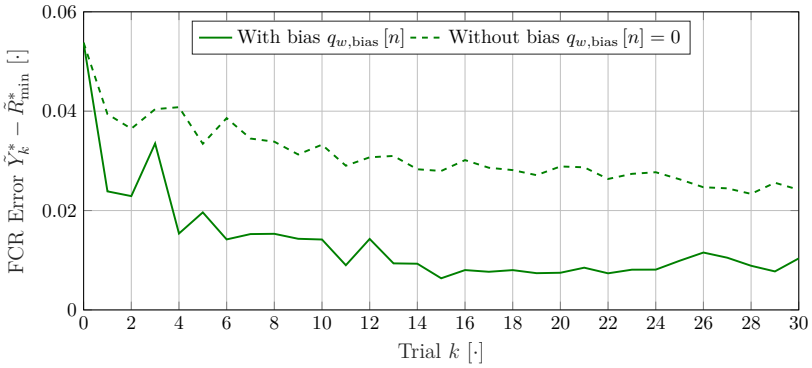
(a) Anchoring.



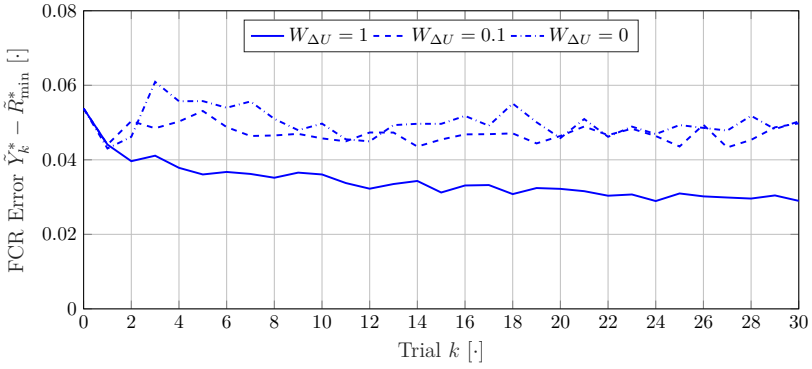
(b) Maximizing Reference.



(c) Anchoring and Maximizing Reference.



(d) D+MR+A with and without weight measurement bias $q_{w,\text{bias}}[n] = 0$.



(e) D+MR+A with different input change cost $W_{\Delta U}$ configurations.

Figure 9: Simulation results – see Section 4 for detailed explanations.

potential batches are selected for the IVS algorithm by maximizing

$$\begin{aligned} \mathcal{B}_k = \arg\max_{\tilde{\mathcal{B}}} \; & N_{\tilde{d}}(\tilde{\mathcal{B}}) \cdot N_{\tilde{y}}(\tilde{\mathcal{B}}) \cdot \min\{\#\tilde{\mathcal{B}}, N_b\} \\ \text{s.t.} \quad & \tilde{\mathcal{B}} \subseteq \{1 - N_{PB}, \ldots, k-1\} \end{aligned} \tag{35}$$

where $\mathcal{B}_k$ is the set of batches used for IVS and training on trial k, $\tilde{\mathcal{B}}$ is a set of potential batch indexes, N_b is the maximum number of batches considered, $N_{\tilde{d}}(\tilde{\mathcal{B}})$ and $N_{\tilde{y}}(\tilde{\mathcal{B}})$ is the number of potential disturbances and outputs with batch indexes $\tilde{\mathcal{B}}$. Moreover, the temperature input, broiler weight output and cumulative feed are required for it to be a potential batch.

Normalized FCR cost function

Batches have different durations, which makes FCR comparison difficult. For this reason, the FCR is normalized to the same weight ψ using the performance measure

$$J_{\text{FCR},\psi}(y_f, y_w) = \frac{y_f\left(1 - \frac{k_w}{\psi}\right) + y_w\left(\frac{k_f}{\psi}\right) - k_f}{y_w - k_w} \tag{36}$$

where $y_f \in \mathbb{R}_+$ is the average feed consumed per broiler, $y_w \in \mathbb{R}_+$ is the average slaughter weight, $\psi = 2.2$ kg, and $k_w = -1.110$ kg and $k_f = -3.081$ kg are correction factors. This expression is created using official regression formulas used by the Danish broiler industry[Det Danske Fjerkæraad, 2013, pp. 85], and replaces the augmented data driven model in (23) by

$$\tilde{Y}_k^*(U) = J_{\text{FCR},\psi}(\tilde{Y}_{k,f}(U), \tilde{Y}_{k,w}(U)). \tag{37}$$

"Extended" TILC

In Figure 10 the terminal system matrix $\hat{\tilde{P}}^*_{k,\kappa_k}$ for $k = 0$ is shown, which has a significant degree of "ripple" from day 21 onwards. This feature is caused by ripples in the training data and falsely suggests that FCR can be decreased by temperature fluctuations, as it results in either cold or heat stress. This promotes a loss of appetite and reduced growth during a period of desired maximum growth, according to the FCR minimization considerations in Section 2.2. A straightforward solution, available within point-to-point ILC framework, is to *extend* the terminal ILC design to include the last $N_\dagger \in \mathbb{Z}_+$ output samples, i.e.,

$$Y_k^\dagger = \begin{bmatrix} y_k^*[N_e - N_\dagger + 1] & \cdots & y_k^*[N_e] \end{bmatrix}^T \in \mathbb{R}_+^{N_\dagger}. \tag{38}$$

The extended ILC problem now is

$$U_{k+1} = \arg\min_{U \in \Omega_{k+1}} \left\| \tilde{E}^\dagger_{\kappa_k}(U) \right\|^2_{W^\dagger_{\tilde{E}}} + \|U - U_{\kappa_k}\|^2_{W_{\Delta U}} \tag{39a}$$

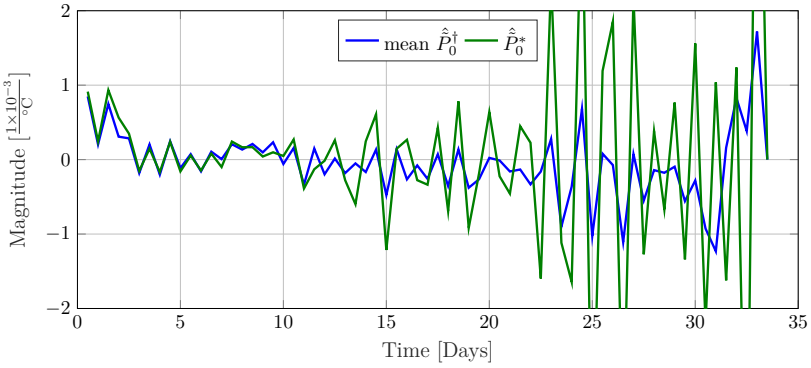


Figure 10: $\hat{\hat{P}}_{k,\kappa_k}$ and the mean $\hat{\hat{P}}^\dagger_{k,\kappa_k}$ for $k = 0$ of the experimental test.

subject to (28),

$$\tilde{R}^\dagger_k = \tilde{Y}^\dagger_k(U_k) - \mathcal{R}^\dagger, \tag{39b}$$

$$\tilde{E}^\dagger_{\kappa_k}(U) = \tilde{R}^\dagger_{\kappa_k} - \tilde{Y}^\dagger_{\kappa_k}(U) \text{ and} \tag{39c}$$

$$\tilde{Y}^\dagger_{\kappa_k}(U) = \hat{\hat{P}}^\dagger_{k,\kappa_k} U + \hat{\hat{K}}^\dagger_{k,\kappa_k} \tag{39d}$$

where $W^\dagger_{\tilde{E}} \in \mathbb{R}^{N_\dagger \times N_\dagger}$, $\tilde{R}^\dagger_k \in \mathbb{R}^{N_\dagger}$, $\hat{\hat{P}}^\dagger_{k,\kappa_k} \in \mathbb{R}^{N_\dagger \times N_\dagger}$. Note that (28) remains unchanged, and this approach is within the point-to-point ILC framework.

5.2 Method Configuration

The input variable selection algorithm selects up to 2 variables from the available disturbances, e.g. CO2 denoted by $d_i[k \mid t]$ with index i, and up to 2 lags are selected per disturbance and input, e.g. $d_i[k-1 \mid t]$ and $d_i[k-3 \mid t]$. The weight shape cost function is configured with N_ϕ = day 15, and the extended TILC is configured with $N_\dagger = 4$ samples. A total of $N_m = 64$ ensemble models are used, of which the remaining settings are identical to the simulation study as described in Section 4.2.

5.3 Experimental Results

Figure 11 shows relevant measured signals for $k = 1$, where the FCR@2.2kg of trial $k = 1$ is approximately 6% smaller compared to $k = 0$. The terminal broiler weight is 200g higher and the terminal cumulative feed consumption is only 100g higher, which is a disproportionate exchange rate. The initial input change is approximately 0.5 °C lower for days 0–4 and 9–15, and approximately 2 °C higher for day 27. The initial decrease in temperature decreased the broiler growth rate, as the operator reported mild signs of cold stress in the broilers on visual inspection.

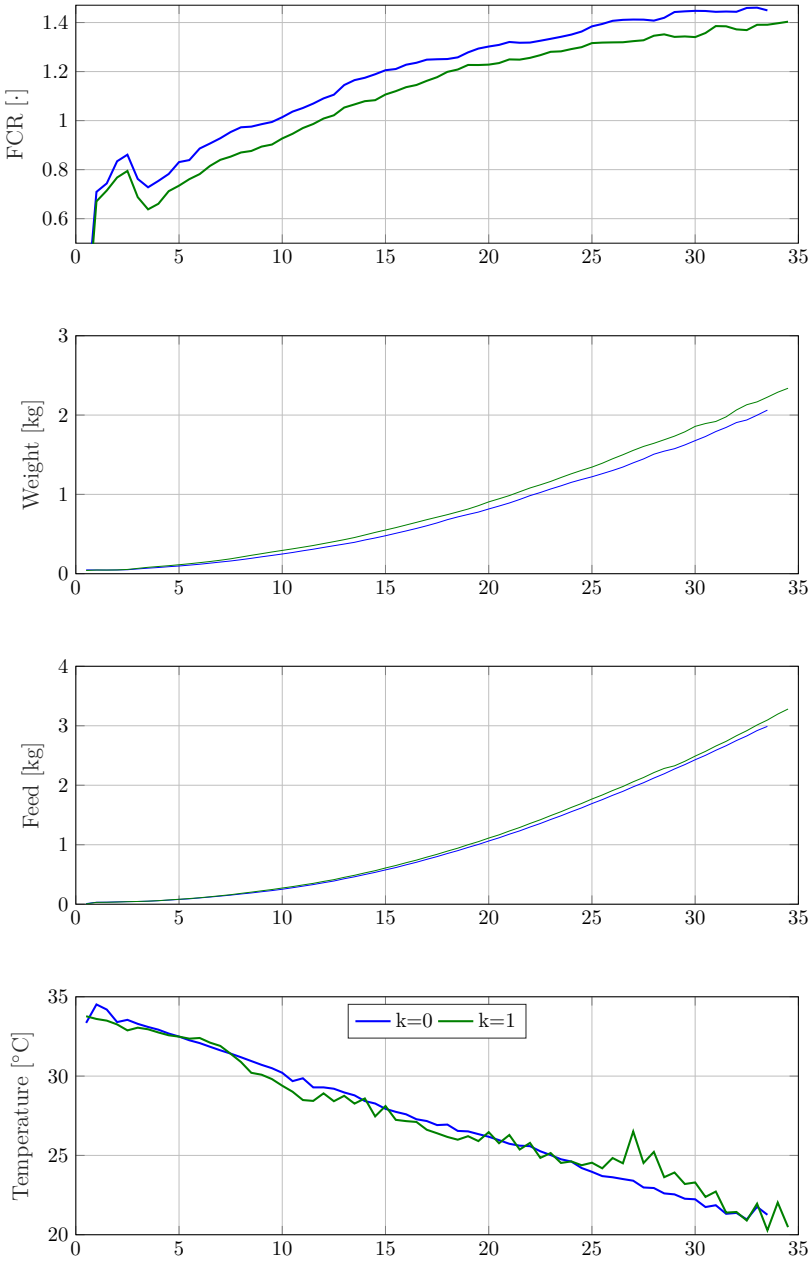


Figure 11: Experimental results for $k = 1$ using the new algorithm. The FCR, broiler weight, feed consumption and measured temperature is depicted for trial $k \in \{0, 1\}$ along with their difference in red.

Applying the new algorithm results in a FCR@2.2kg decrease of 5.9% (0.059) and FCR decrease of 1.4% (0.014) for trial $k = 1$ – calculated using the slaughter weight. In Figure 12 the historic performance of the house is given, which shows that trial $k = 1$ has a very promising historically low FCR. This result is very close to the trial-to-trial FCR decrease for the first trial of the simulation study in Figure 9 with a FCR decrease of approximately 1% (0.01).

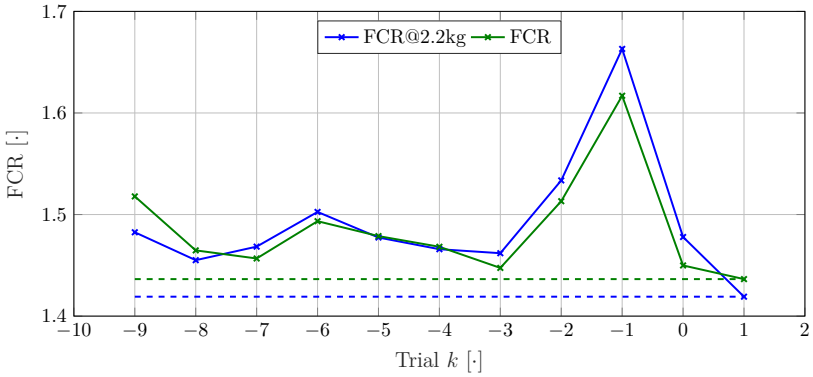


Figure 12: FCR and FCR@2.2kg performance overview of the recent 10 trials $k \in \{-9, \ldots, 0\}$ and the current trial $k = 1$. Trial $k \in \{-2, -3\}$ have unusually high FCR due to an unusually cold winter, rendering the temperature regulation unable to maintain the desired temperature.

These experimental results demonstrate the basic feasibility of the new algorithm and provide a basis for onward development. One obvious area is to verify the long term results of the algorithm, as sentient biological production can sometimes give misleading results.

6 Conclusions and Future Work

In this paper a heuristic broiler growth model has been formulated and used to investigate the performance of a data driven feed conversion rate (FCR) optimization based ILC law in simulation and in practice. Traditional ILC is modified to minimize the terminal broiler FCR and to better cope with the uncertain nature of the data driven model. The heuristic broiler growth model is based on experience from a broiler application expert and approximates the dynamic behavior between broiler weight, feed uptake and temperature, including a measurement weight bias commonly known to exist in the *state-of-the-art* broiler production. Extensive simulation based studies confirm the potential of this approach, but the measurement weight bias is found to reduce the trial-to-trial convergence rate. The simulation study notably showed that iterative learning is required for FCR minimization. Further modifications were

made to prepare the algorithm for experimental testing on a real broiler house, and a FCR reduction of 1.4% was obtained over a single operation in a broiler house with around 40,000 broilers. Areas for future work include studying the long term properties of this algorithm as mentioned in the last section and decreasing the effects of the measurement weight bias.

References

H.-S. Ahn, Y. Chen, and K. Moore. "Iterative learning control: Brief survey and categorization". *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1099 –1121, 2007.

S. Arimoto, S. Kawamura, and F. Miyazaki. "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems". In *The 23rd IEEE Conference on Decision and Control*. IEEE, dec 1984. doi:10.1109/cdc.1984.272176.

Aviagen. *Ross 308 Broiler: Performance Objectives*, 2014. URL: http://en.aviagen.com/assets/Tech_Center/Ross_Broiler/Ross-308-Broiler-PO-2014-EN.pdf.

D. A. Bristow, M. Tharayil, and A. G. Alleyne. "A survey of iterative learning control". *Control Systems, IEEE*, 26(3):96 – 114, 2006.

B. Chu, C. T. Freeman, and D. H. Owens. "A novel design framework for point-to-point ILC using successive projection". *IEEE Transactions on Control Systems Technology*, 23(3):1156–1163, may 2015. doi:10.1109/tcst.2014.2356931.

Det Danske Fjerkæraad. *Årsberetning 2013*. 2013. https://danskfjerkrae.dk/om-fjerkraebranchen/det-danske-fjaerkraeraad/aarsberetning/aarsberetning-2013.

K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer Science + Business Media, 2014. doi:10.1007/978-1-4471-5571-3.

C. T. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes, and K. L. Meadmore. *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation*. Springer, 2015. ISBN 978-1-4471-6725-9. doi:10.1007/978-1-4471-6726-6.

J. Grahovac, A. Jokić, J. Dodić, D. Vučurović, and S. Dodić. "Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks". *Renewable Energy*, 85:953–958, jan 2016. doi:10.1016/j.renene.2015.07.054.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Data driven broiler weight forecasting using dynamic neural network models". *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. doi:10.1016/j.ifacol.2017.08.1073.

S. V. Johansen, M. R. Jensen, B. Chu, J. D. Bendtsen, and E. Rogers. "Broiler growth optimization using norm optimal terminal iterative learning control". *Proceedings of Control Technology and Applications*, 2018.

S. V. Johansen, J. D. Bendtsen, and J. Mogensen. "Broiler slaughter weight forecasting using dynamic neural network models". *Proceedings of the International Conference on Industrial Engineering and Applications*, 2019a. doi:10.1109/IEA.2019.8714850.

S. V. Johansen, J. D. Bendtsen, M. Riisgaard-Jensen, and J. Mogensen. "Broiler weight forecasting using dynamic neural network models with input variable selection". *Journal of Computers and Electronics in Agriculture*, 2019b. doi:10.1016/j.compag.2018.12.014.

I. Lim, D. J. Hoelzle, and K. L. Barton. "A multi-objective iterative learning control approach for additive manufacturing applications". *Control Engineering Practice*, 64(11):74–87, 2017.

R. Nasimi and R. Irani. "Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks". *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14):1604–1611, may 2014. doi:10.1080/15567036.2011.592903.

OECD. *OECD-FAO Agricultural Outlook 2018-2027*. OECD Publishing, jul 2018. doi:10.1787/agr_outlook-2018-en.

W. Paszke, E. Rogers, K. Galkowski, and Z. Cai. "Robust finite frequency range iterative learning control design and experimental verification". *Control Engineering Practice*, 21(10):1310–1320, 2013.

D. Prichard and J. Theiler. "Generating surrogate data for time series with several simultaneously measured variables". *Physical Review Letters*, 73(7):951–954, aug 1994. doi:10.1103/physrevlett.73.951.

H. sung Ahn, K. Moore, and Y. Chen. "LMI approach to iterative learning control design". jul 2006. doi:10.1109/smcals.2006.250694.

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-511-6

AALBORG UNIVERSITY PRESS